The University of Queensland

Bachelor of Engineering Thesis

Thrust Vector Control and Guidance of a Flyback Booster during Launch

Student Name: Tasarinan JOUIR

Course Code: MECH4500

Supervisor: Michael Smart

Submission Date: 25 October 2017

A thesis submitted in partial fulfilment of the requirements of the

Bachelor of Engineering Degree in Mechanical Engineering

UQ Engineering

Faculty of Engineering, Architecture and Information Technology

[This page has been intentionally left blank.]

Abstract

Throughout its short history, the space industry has been a subject to high levels of secrecy – initially due to concerns for national security, and more recently as a result of limitations imposed by commercial intellectual property. Fundamental to a successful rocket launch, a comprehensive understanding of thrust vector control and guidance techniques are required to enable effective first stage system design. This shortfall in the technical literature is tackled through investigation of existing methodologies, with each selected algorithm derived from first principles and simulated using the aerospace software package, CADAC++ [1]. The modelling campaign was tailored to meet the requirements of a conceptual three-stage rocket-scramjet-rocket system [2], however the fully characterised generic Small Launch Vehicle model [3] was used for simulation purposes. Results are presented demonstrating the performance of a multiple-booster thrust vector control model. Guidance simulation attempts were unsuccessful, however an analysis of the null result is presented, accompanied by recommendations for an improved simulation architecture. Fundamentally, this work culminates the body of knowledge required for the design of a first-stage guidance module based on multiple-booster thrust vector control.

Contents

List of Figures

List of Tables

1 Introduction

Advances in satellite technology along with renewed aspirations for human space exploration have motivated the development of increasingly efficient launch systems. Classically, these have consisted of multi-stage rockets, however it is theorised that systems comprised of rocket-scramjet-rocket stacks will provide higher efficiency due to the lower fuel-oxidiser weight requirements associated with air-breathing engines. Both approaches require knowledge of the first-stage boost engine cut-off (BECO) in order to enable efficient second-stage design (this is true of any multi-stage system using a first-stage rocket booster). Implicitly this translates to the requirement that a launch system must be able to achieve a particular flight condition before the initiation of the second-stage. When considering systems with a scramjet second-stage, achieving the desired BECO state is of even more importance due to the limited operational window of the air-breathing engine.

The dynamic information required to transition the launch system from the ground to the desired BECO state is provided by the guidance system. Specifically, the guidance system generates a command at each time step (i.e. an acceleration vector increment or direction change) such that the vehicle will achieve the desired end state subject to a set of restrictions (such as a maximum time or fuel expenditure). While guidance information is critical for a successful launch, it is useless unless the system can react accordingly. For first-stage boosters, the action that the launch system takes is typically changing the direction of the thrust vector. Control fins can be used to provide an aerodynamic control, however they are not considered in this study as they are less efficient [8] and industry has moved away from using them. Thrust vector control is defined here as the process of converting an incremental guidance command to a change in the thrust vector direction. The most efficient method of altering the direction of the thrust vector for liquid fuel rockets is through the independent gimballing of each booster engine, so the change in thrust vector direction is typically achieved via an actuator deflection. Alternative approaches applicable to solid rockets (i.e nozzle gimballing)

or methods involving exhaust manipulation (such as deflection via secondary asymmetric fuel injection or control fins within the exhaust flow) are also not considered in this study.

While multi-stage launch systems have been in active service since the space race, state-of-the-art guidance and thrust vector control strategies are often not released into the public domain due to the secretive nature of the space industry (arising historically from national security issues and as a result of the value associated with commercial intellectual property in contemporary times). This study aims to remedy this shortfall by investigating what the state-of-the-art strategies are, and providing a comprehensive approach for the design of a first-stage guidance algorithm to achieve a desired BECO state, including the design of a thrust vector control strategy using independently gimballed booster engines. Performance evaluation will be completed through the implementation of the strategy in a 6 degree of freedom (6DOF) launch simulation. Of particular interest to the authors is the proposed SPARTAN rocket-scramjet-rocket launch system [2], and as such that vehicle along with its design trajectory will be taken as the primary case study. The SPARTAN first stage consists of a cluster made up of three fly-back boosters, so efforts will be made to independently validate any methods applicable to a single booster prior to the full vehicle integration. Note that contrary to the modelling procedure, the structural, propulsive and aerodynamic properties used during the simulation process were based on the Small Launch Vehicle used by Zipfel [3].

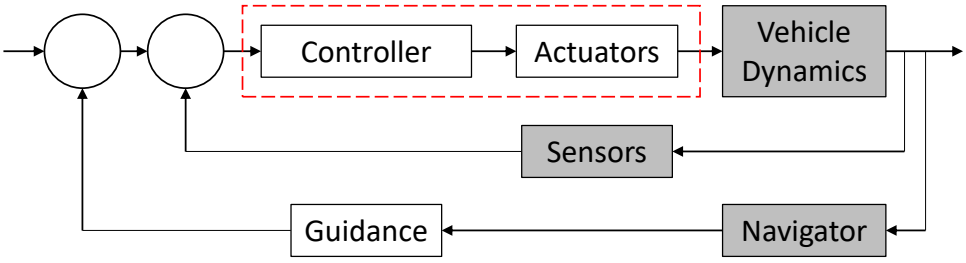

Figure 1: The relationship between guidance, navigation and control (adapted from [4]). Note that for this study, thrust vector control is defined as including both the controller and actuators as outlined in dashed red. Components in grey are included in the simulation, however are not considered independently.

A graphical representation of the scope of this study is shown in Fig. 1, which

illustrates the relationship between guidance, navigation and control in an aerospace context. Of interest are the components in white, noting that the pair grouped within the red dashed outline are defined together as thrust vector control. Note also that this study assumes that all sensors are ideal and that the navigator is simply represented by the pre-specified end state.

This manuscript is organised as follows: Chapter 2 provides a review of thrust vector control and guidance literature, including both controller and actuator models as well as explicit guidance schemes. Chapters 3-5 outline the modelling approach, with Chapter 3 describing the underlying coordinate systems, equations of motion and vehicle models. Chapter 4 provides first-principles derivations of both the single and multiple booster models, an actuator model as well as acceleration and roll controllers. Chapter 5 considers first a planar variation of linear tangent guidance, before extending the method into 6 degrees of freedom (6DOF) using a vector based formulation. Chapter 6 initially provides an overview of the simulation architecture, then lists simulation results for both the thrust vector control and guidance algorithms. Finally, Chapter 7 consists of a discussion on the performance of each algorithm, including all identified limitations and suggestions for future work.

2 Literature Review

Thrust vector control (TVC), by definition, provides angular control of a rocket's trajectory through simple deflection of the thrust vector. In the fundamental case of a single booster system, the gimballed engine can be rotated through two degrees of freedom. The resultant torques when deflecting a single booster provide sufficient fidelity for both pitch and yaw control. For systems with multiple boosters, pitch and yaw moments are attained by symmetrically deflecting all boosters. Additional boosters inherently provide additional thrust vectors which are offset from the longitudinal body axis, incorporating a further degree of rotational freedom. This allows for the roll angle to be controlled using smaller, differential angular deflections [9].

Mechanically, TVC traditionally uses an orthogonal pair of actuators for each booster (typically hydraulic, electro-hydraulic or electro-mechanical). Space X, a leading private launch company, utilise hydraulic TVC actuators for their Merlin engines (see Fig. 2A). These operate using high pressure kerosene provided via the booster turbopump, removing the risk of TVC failure due to insufficient hydraulic fuel. In addition, the Merlin engine turbopump exhaust nozzle can also be deflected to assist with roll control [10]. A more recent entrant into the space industry, Rocket Lab, use electromechanical actuators to gimbal their Rutherford engines (see Fig. 2B) [11].

While these actuators provide the physical mechanism for TVC, and thus need to be included in any effective modelling scheme, additional control strategies are required to convert commands from the vehicle's guidance system into the required angular deflection commands. An account by Orr et al. [12] on NASA's Space Launch System (SLS) highlights the organisation's practise of relying on proven methods for TVC. This translates to ensuring the flight control system dynamics are linearisable, as well as using classical proportional-integral-derivative (PID) control and gain scheduling. As a result, robust evaluation of system performance is possible using classical frequency-domain stability margin criteria. Due to the large size of the SLS, that study also investigates the

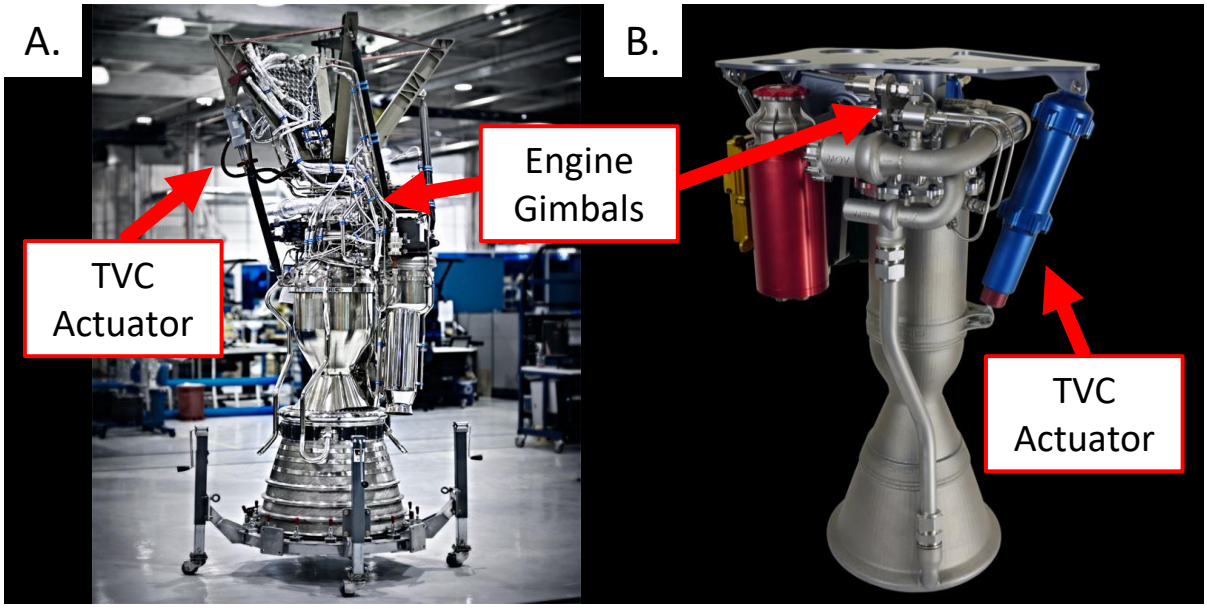

Figure 2: A: The SpaceX Merlin engine, featuring hydraulic TVC actuators (image: [5]). B: The Rocket Lab Rutherford engine, featuring electromechanical TVC actuators (image: [6]).

effects of fuel slosh and structural bending. Finally, Orr et al. emphasise the importance of being able to enhance TVC capability on demand and maximise robustness to failures (considering either a single engine or sensor failure). Ensworth [13] presents a NASA study which also models TVC for a three booster nuclear thermal rocket cluster using PID controllers. That study considers the affect of engine asymmetries and fuel slosh, while also using a reaction control system (RCS) for roll control. A similar policy of utilizing proven control techniques to enable higher complexity guidance algorithms is employed by SpaceX (information about Space X was gathered from informal sources due to limited official statements). Zipfel [4] describes a series of such controllers for general aerospace vehicles including a roll position autopilot and an acceleration control scheme developed for agile missiles. That book also derives a flight-path-angle controller for aircraft climb and descent. While this may be the current industry approach, advanced control algorithms have been developed to adapt to non-catastrophic failures through in-flight model reconstruction of the failed system using a neural network [14]. A study by Shtelle et al. [15] presents a thrust vector control scheme which minimises the effect

of disturbances such as wind gusts through the use of sliding mode control, while also replacing the need for the classically required gain scheduling. Yu and Shu [16] present a compound control law which combines sliding mode control and PID control with feed-forward compensation to improve the transient and position tracking performance of the thrust vector control system.

Building accurate system models is an integral part of control systems design. In the case of thrust vector control (TVC), this includes developing representations of actuators, sensors and vehicle dynamics. Schinstock et al. [17] present results using a simple two mass-spring-damper system to capture both the actuator and engine moments of inertia, as well as actuator and gimbal friction. Lazic et al. [18] approach the problem by employing an open-loop transfer function to represent the hydraulic actuators and also notably use angular transducers for position feedback rather than direct measurement of actuator extension. Both studies validate their respective approaches using the equivalent control problem of an inverted pendulum-actuator experiment. A study by Li et. al [19] uses a 2 degree of freedom model to represent an electromechanical actuator, consisting of two sequential transfer functions to convert from a voltage command input to the gimbal angle. The model presented in that study considers the affect of actuator stiffness, both engine and motor inertias, as well as the internal motor gear ratio. Thrust vectoring was achieved through a compound controller utilizing either PID or bang-bang control based on a measure of actuator linear displacement error. Zipfel [4] presents an approach which simply represents actuator dynamics using a second order spring-mass-damper system combined with deflection and deflection rate limiters. Linearisable models such as these enable classical control system design techniques to be applied directly, allowing for the transient and steady state response to be treated using parameters such as rise-time, settling time and percent overshoot [20].

Weight considerations are fundamental in launch vehicle design, with the Tsiolkovsky rocket equation underpinning the core relationship between payload mass and fuel requirements given a structural capability. With the emphasis this places on fuel

efficiency, it is of no surprise that the key design objective for trajectory optimisation is fuel conservation [21]. Suresh [22] notes that to achieve such an objective, there exist two distinct design paradigms for rocket guidance: a) implicit, by which the vehicle tracks a pre-determined trajectory; and explicit, where a steering law is applied during flight to determine the actions required to achieve a specified end state. Inherently, explicit schemes have larger computational requirements associated with the need to iteratively update the steering law during flight. However, as the exact nature of the trajectory is not required prior to take-off, explicit schemes are more robust to deviations from the desired path. Dual mode guidance algorithms take advantage of both schemes, with the vehicle following a trajectory generated by an explicit approach over larger time steps. This study will only consider explicit schemes.

The cross product steering law is an explicit scheme shown to provide a near fuel optimal solution for non-constant gravitational fields [23]. That approach simply attempts to match the velocity vector of the vehicle with a final-state velocity vector requirement by minimising the difference between those quantities, termed the velocity-to-go. The minimisation is optimally enforced through recognition that aligning the velocity-to-go vector with its time derivative drives it to zero, and that they are aligned when the cross product of the two vectors is equal to zero [24].

Teren [25] presents an explicit scheme based on an approximate closed form solution of the equations of motion. The resulting formulation uses both pitch and yaw steering laws to achieve a final state (altitude, velocity and flight angle), while incorporating a pseudotarget methodology to assist in reducing the error introduced through assumptions. That study assumes a spherical Earth, an inverse-square law for gravity and constant thrust.

The bilinear tangent law was first described by Lawden [26] as a method of obtaining a fuel optimal rocket trajectory solution. The bilinear tangent law is based on the calculus of variations, building on recognition that an assumption of constant thrust makes a time optimal solution equivalent to a fuel optimal one. The additional restrictive

assumptions of constant thrust and gravity as well as no aerodynamics are also made. These are of little consequence however if the steering law is iteratively updated with high fidelity environmental models in place. A planar derivation of the law is described by Bryson and Ho [27], who take a classical calculus of variations perspective on the problem. A full algorithm derivation and suggested implementation methodology of the guidance scheme in 6DOF is outlined by Jaggers [28], who identifies the caveat that the algorithm becomes unstable as the time-to-go approaches zero. That variation was used as the steering law for the Space Shuttle's Powered Explicit Guidance (PEG) scheme. Zipfel [3] presents a method of simulating the strategy for a three-stage launch using the software package CADAC++ [1].

Sinha and Shrivastava [29] describe an optimal explicit guidance scheme for a multi-stage vehicle with emphasis on increasing the capability for high yaw manoeuvres (a necessity for launch sites located away from the equator). That study also presents methodology for incorporating variations in both thrust and gravity. An active guidance algorithm for finless rockets is presented by Gomez and Miikkulainen [8]. Finless rockets have increased efficiency over their finned counterparts, at the cost of reduced passive stability control. Gomez and Miikkulainen utilize a neural network to solve the resultant non-linear control problem, linking the rocket state to differential thrust commands. An excellent, more extensive, overview of guidance literature, including implicit methods, is presented by Robinson [21].

This study adheres to the design philosophy utilised by NASA and SpaceX of applying simple TVC to enable sophisticated guidance. Thrust vector control is modelled for both single and multiple booster cluster configurations, using mass-spring-dampers to represent actuator dynamics. Acceleration controllers (as well as a roll controller for the multiple booster case) are used to convert guidance commands to changes in the thrust vector direction. Each algorithm is derived from first principles then applied to a simulated first stage launch using the CADAC++ software package.

3 System Model

Two vehicle models are considered throughout this study, the first: the three stage rocket-scramjet-rocket conceptual launch system [2], and the second: the generic Small Launch Vehicle used by Zipfel [3]. As the rocket-scramjet-rocket system is still under development and is not yet fully characterised, that vehicle is only addressed theoretically. Consequentially, the Small Launch Vehicle model is used in this study for all simulation purposes, however the control and guidance algorithms were selected to meet the requirements of the rocket-scramjet-rocket system. The coordinate systems and general mathematical notation used throughout this manuscript are described in Section 3.1, along with the governing equations of motion in Section 3.2. A qualitative description of the rocket-scramjet-rocket system first stage is provided in Section 3.3, including an untested theoretical aerodynamic model. Finally, the Small Launch Vehicle is characterised in Section 3.4. The proposed first-stage boost engine cut off (BECO) state for the rocket-scramjet-rocket system as investigated by Chai [7] is used as a baseline during simulations (see Tab. 1).

Velocity (m/s)	Mach	Altitude (m)	$\bar{q}$ (kPa)	t_{burn} (s)
1517.9	5.1	24094	53	130

Table 1: SPARTAN first-stage burnout state [7].

3.1 Coordinate Systems

The mathematical notation developed by Zipfel [4] is adopted here, with lower case and upper case square bracketed variables indicating vectors and matrices respectively. An additional subtlety is introduced through the use of sub and superscripts. Superscripts placed on variables (inside the square brackets) indicate the frame of reference, while subscripts denote a point of reference. Additionally, superscripts attached to the square brackets directly (i.e. $]^I$) indicate the coordinate system. Transformation matrices are exclusively represented by the capital letter T, with double superscripted square brackets

indicating the direction of the transform – from the right-hand superscript to the left. A sequence of transformation matrices can be used with ease if organised correctly. For example, a vector $[x]^A$ can be converted from coordinates $]^A$ to coordinates $]^C$, given the transformation matrices $[T]^{BA}$ and $[T]^{CB}$, through the computational sequence $[x]^C = [T]^{CB}[T]^{BA}[x]^A$. Finally, an overbar ($^-$) represents the transpose of a vector or matrix, with an overbarred transformation matrix reversing the direction of the transformation. Seven coordinate systems are considered in this study, again based on the definitions presented by Zipfel [4]:

1. Inertial Coordinates, $]^I$.
2. Earth Coordinates, $]^E$.
3. Geographic Coordinates, $]^G$.
4. Body Coordinates, $]^B$.
5. Stability Coordinates, $]^S$.
6. Aeroballistic Coordinates, $]^R$.
7. Engine Gimbal Coordinates, $]^N$.

The first six coordinate systems are defined in detail here, along with the associated angles and transformation matrices. Engine gimbal coordinates are described in Section 3.1, where they are fundamental in enabling thrust vector control. The notation used to indicate the three coordinate axis for a given coordinate system $]^C$, is 1^C, 2^C and 3^C.

3.1.1 Inertial Coordinate System

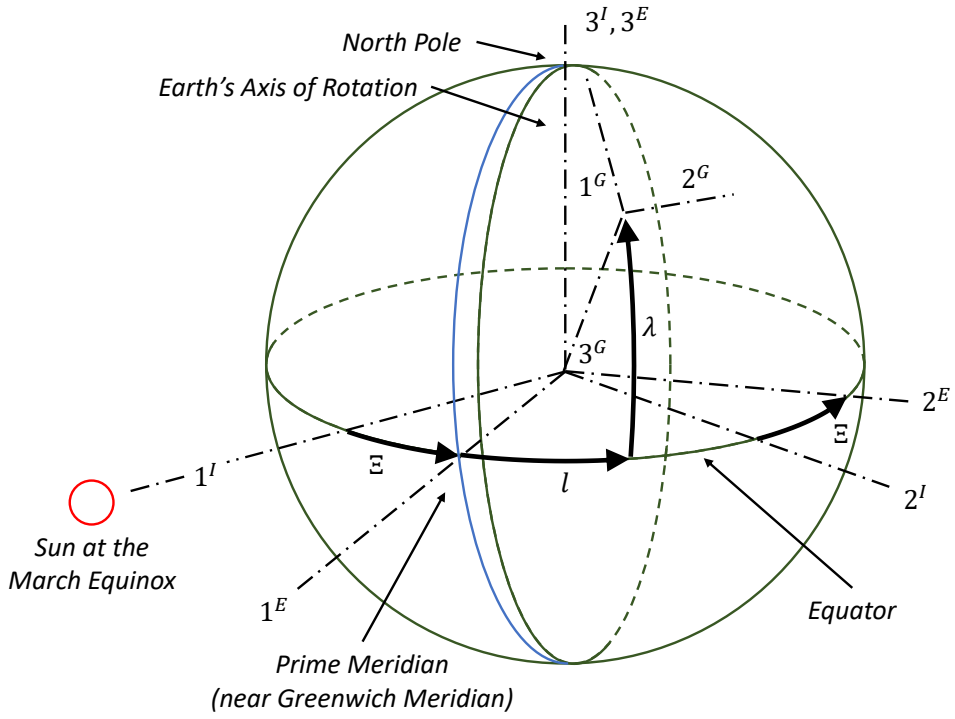

Figure 3: The three Earth centred coordinate systems, consisting of inertial ($]^I$), Earth ($]^E$) and geographic coordinates ($]^G$).

Inertial coordinates form the base Earth centred coordinate system used in this study. The 1^I axis is defined such that it passes through the vernal equinox, while the 3^I axis passes through the north pole. (The vernal equinox is the point on the equator that intersects the ground track made by the sun on the Earth at the march equinox. The place on the equator varies from year to year, but represents a relatively fixed relationship of the earth to the sun and the surrounding stars.) The 2^I axis completes the right hand coordinate system, passing through the equator.

3.1.2 Earth Coordinate System

The second Earth centred coordinate system, known as the Earth coordinate system, is not used directly in this study. It is included here as it is necessary for conversion between geographic and inertial coordinates. The 1^E axis is defined as the point where the Prime Meridian (near to, but more precise than the Greenwich Meridian) passes through the

equator, while the 3^E axis coincides with the spin axis of the Earth. Similar to inertial coordinates, the 2^E axis passes through the equator to complete the right hand coordinate system. Noting that the 3^I and 3^E axis are coaxial, the transformation between the two coordinate systems can be defined in terms of a single angle, known as the hour angle (Ξ). The hour angle subtends the arc created between the Prime Meridian and the vernal equinox. The resultant transformation matrix is defined as:

$$[T]^{EI} = \begin{bmatrix} \cos\Xi & \sin\Xi & 0 \\ -\sin\Xi & \cos\Xi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

3.1.3 Geographic Coordinate System

The geographic coordinate system is constructed such that it sits on the surface of the Earth, and is defined with respect to the Earth coordinate system using the familiar longitude and latitude angles (denoted by l and λ respectively). The geographical coordinate system is the natural system for representing gravitational forces. The 3^G axis is defined such that it points to the centre of the Earth, along with the 1^G axis pointing north and the 2^G axis east. The transformation matrix from Earth to geographic coordinates is given by:

$$[T]^{GE} = \begin{bmatrix} -\sin\lambda\cos l & -\sin\lambda\sin l & \cos\lambda \\ -\sin l & \cos l & 0 \\ -\cos\lambda\cos l & -\cos\lambda\sin l & -\sin\lambda \end{bmatrix} \tag{2}$$

3.1.4 Body Coordinate System

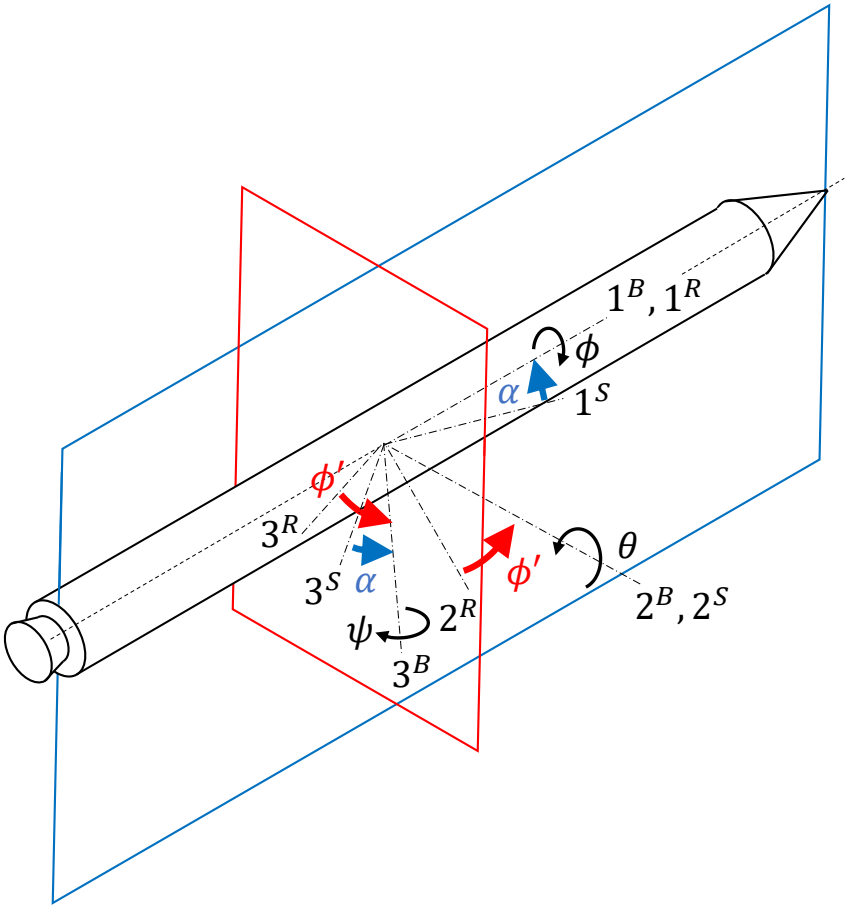

Figure 4: The three vehicle centred coordinate systems, consisting of body ($]^B$), stability ($]^S$) and aeroballistic coordinates ($]^R$). The symmetry and load factor planes of the vehicle are emphasised in blue and red respectively, with axis rotations coloured to match the parent plane.

The body coordinate system is the first of the vehicle centred systems used in this study, chosen for its convenience when considering body forces. The 1^B axis is defined such that it points through the nose of the vehicle. The 3^B axis is then taken to point downward through the plane of symmetry (arbitrarily chosen for axisymmetric vehicles), while the 2^B axis passes through the right wing (or equivalent) to complete the right hand coordinate system. The angular displacement with respect to the geographical coordinate system is described by the Euler angles for roll (ϕ), pitch (θ) and yaw (ψ).

The corresponding transfer function is:

$$[T]^{BG} = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\cos\phi & \cos\theta\cos\phi \end{bmatrix} \quad (3)$$

3.1.5 Stability Coordinate System

Stability coordinates are a vehicle centred coordinate system used to conveniently describe aerodynamic effects. The stability coordinate system is related to body coordinates through a rotation about the shared 2 axis, equal to the angle of attack (α). The 1^S axis then exists within the symmetry plane of the vehicle and further, aligns with the vehicle's velocity vector in that plane. The 3^S axis is placed as required to complete the right hand coordinate system. The transfer function from stability to body coordinates in terms of the angle of attack is given by:

$$[T]^{BS} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad (4)$$

3.1.6 Aeroballistic Coordinate System

Aeroballistic coordinates are a vehicle centred coordinate system used for rotationally symmetric vehicles, differing from stability coordinates by placing emphasis on the load factor plane. Aerobalistic coordinates are considered here as they are used by Zipfel to describe the aerodynamics of the Small Launch Vehicle. The 1^R axis coincides with the 1^B axis, with the right hand body coordinate system then rotated about that axis by the aerodynamic roll angle ϕ' to obtain the orientation of the aeroballistic coordinates. The

transfer function from aeroballistic coordinates to body coordinates is given by:

$$[T]^{BR} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi' & \sin\phi' \\ 0 & -\sin\phi' & \cos\phi' \end{bmatrix} \tag{5}$$

3.2 Equations of Motion

The equations of motion described by Zipfel [4] are restated below, using the notation outlined in Section 3.1. The translational equations of motion are given by Eq. 6:

$$\left[\frac{dv_B^I}{dt}\right]^I = \frac{1}{m}[\bar{T}]^{BI}[f_{a,p}]^B + [\bar{T}]^{GI}[g]^G \tag{6}$$

Where v_B^I is the velocity of the vehicle with respect to the inertial frame, $f_{a,p}$ represents the aerodynamic and propulsive forces and g the gravitational acceleration. Note the use of inertial $(]^I)$, body $(]^B)$ and geographic $(]^G)$ coordinates along with the respective transformation matrices. Similarly, the rotational equations of motion are:

$$\left[\frac{d\omega^{BI}}{dt}\right]^B = \left([I_B^B]^B\right)^{-1}\left(-[\Omega^{BI}]^B[I_B^B]^B[\omega^{BI}]^B + [m_B]^B\right) \tag{7}$$

Where ω is the angular velocity of the vehicle, I is the moment of inertia and m_B represents the aerodynamic and propulsive moments about the vehicle's centre of mass, B. Note that here Ω^{BI} denotes the angular velocity of the body frame with respect to the inertial frame.

3.3 SPARTAN First-Stage

The proposed first-stage of the rocket-scramjet-rocket system utilises a cluster configuration consisting of three parallel boosters. This enables complete 3 degree of freedom (3DOF) thrust vector control (TVC), and forms the basis of the multiple booster TVC model developed in this study. The consequence of such a configuration is that the resul-

tant "flat" body shape must be considered within the aerodynamic model (see Section 3.3.1). Noting that the flat shape could be used to generate some lift, potentially aiding manoeuvres, future work will consider the optimal roll orientation for a first-stage launch. The fly-back boosters used as a reference in this study are based on the "Sprite" pods by Microcosm [30], with adaptations as described by Chai [7]. The booster specifications are listed in Tab. 2.

Thrust, sea (N)	Thrust, vac (N)	I_{sp}, sea (s)	I_{sp}, vac (s)	Mass (kg)
74730	88960	239	285	5127.99

Table 2: Fly-back booster properties [7].

Note that although each model in this study was chosen while considering the SPARTAN first-stage as the reference design, all simulations were completed using the fully characterised Small Launch Vehicle model outlined in Section 3.4.

3.3.1 Aerodynamic Model

The aerodynamic model derived here is based on the formulation used by Zipfel [4] to represent a hypersonic vehicle, however includes adjustments to make it suitable for use with rocket aerodecks. Specifically this requires using a reference area (S) and length (diameter, d) rather than a reference area, chord and span, as well as the exclusion of all control surface terms. An important caveat of the model presented here is that it is purely theoretical and untested in a simulation environment. By expressing the aerodynamic forces in stability axes, the following equation is obtained:

$$[f_a]^S = \bar{q}S \begin{bmatrix} -C_D \\ C_Y \\ -C_L \end{bmatrix} \tag{8}$$

With C_D indicating the drag coefficient, C_Y the side force coefficient and C_L the coefficient of lift. The dynamic pressure is represented by $\bar{q}$. Similarly, the aerodynamic moments

are expressed in body axes:

$$[m_B]^B = \bar{q}Sd \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} \tag{9}$$

Where C_l, C_m and C_n are the rolling, pitching and yawing moment coefficients respectively. Taking the Taylor-series expansion of the force equation, while considering only linear terms and disregarding all control surface terms, provides expressions for the force coefficients:

$$C_L = C_{L_0}(M,\alpha) + C_{L_\alpha}(M,\alpha)\alpha + C_{L_q}(M,\alpha)\frac{qd}{2V} \tag{10}$$

$$C_D = C_{D_0}(M,\alpha) + C_{D_\alpha}(M,\alpha)\alpha \tag{11}$$

$$C_Y = C_{Y_0}(M,\alpha) + C_{Y_\beta}(M,\alpha)\beta + C_{Y_p}(M,\alpha)\frac{pd}{2V} + C_{Y_r}(M,\alpha)\frac{rd}{2V} \tag{12}$$

In each above equations, terms subscripted by 0 represent a reference value of that coefficient. The variables p, q and r represent the roll, pitch and yaw rates respectively, while V denotes the vehicle velocity magnitude. Furthermore, when considering terms with an angular rate (p, q or r) as the secondary subscript (i.e. C_{L_q}), the coefficient represents the impact of that rate on the first subscript (in the prior example, the effect of the pitch rate, q, on lift, L). The remaining terms are defined as: C_{L_α} - the lift slope derivative; C_{D_α} - the drag derivative, modelling the change in drag force due to the angle of attack; and C_{Y_β}, known as the weather-vane derivative with respect to the side-slip angle. Following a similar analytic process with the moment equation generates the set of moment coefficients:

$$C_m = C_{m_0}(M,\alpha) + C_{m_\alpha}(M,\alpha)\alpha + C_{m_q}(M,\alpha)\frac{qd}{2V} \tag{13}$$

$$C_l = C_{l_0}(M,\alpha) + C_{l_\beta}(M,\alpha)\beta + C_{l_p}(M,\alpha)\frac{pd}{2V} + C_{l_r}(M,\alpha)\frac{rd}{2V} \tag{14}$$

$$C_n = C_{n_0}(M,\alpha) + C_{n_\beta}(M,\alpha)\beta + C_{n_p}(M,\alpha)\frac{pd}{2V} + C_{n_r}(M,\alpha)\frac{rd}{2V} \tag{15}$$

Where C_{m_α} is the derivative representing the pitch moment due to the angle of attack, C_{n_β} is the yaw moment derivative and C_{l_β} represents the dutch-roll derivative, coupling pitch and yaw for a lifting body. The remainder of the terms in the moment coefficient equations follow the double subscript rate derivative or reference value conventions outlined above.

3.4 Small Launch Vehicle

The vehicle model used throughout the simulation process is the generic three-stage, single-booster Small Launch Vehicle presented by Zipfel [3]. As the focus of this study is the first stage, the model reproduced here only considers the vehicle prior to the second-stage separation. The propulsive characteristics of the vehicle are described in Tab. 3 while the structural properties are listed in Tab. 4.

Thrust, vac (N)	I_{sp}, vac (s)	Exit Area (m^2)	Mass (kg)
1407610	279.2	1.0601	48983.7

Table 3: Small Launch Vehicle propulsion characteristics [3].

x_{cg} (m)	I_{xx} (kgm^2)	I_{yy} (kgm^2)	I_{zz} (kgm^2)	L (m)	D (m)
10.5265	21943.38	671626.02	671624.75	16.84	2.032

Table 4: Small Launch Vehicle structural properties [3].

Zipfel [3] uses an aerodynamic model developed for tetragonally symmetric missiles, with all control surface derivatives set to zero, for his Rocket6 simulation. A single dash is used here to indicate the formulation in aeroballistic coordinates (see Section 3.1.6), with the axial and normal force coefficients given by:

$$C'_A = C_{A_0}(M) + C_{A_\alpha}(M)\alpha' + C_t(M) \tag{16}$$

$$C'_N = C_{N_0}(M, \alpha') \tag{17}$$

Where M indicates the mach number and α' the total angle of attack. C_t represents a correction factor for thrust induced drag. The single moment coefficient considered in

aeroballistic coordinates is:

$$C'_m = C_{m_0}(M, \alpha') + C_{m'_q}(M)\frac{q'd}{2V} - C'_N(x_{cg,ref} - x_{cg}) \tag{18}$$

With V representing the vehicle velocity magnitude, q' the body rate in the manoeuvre plane (pitch, yaw or a combination of the two) and d the reference length. All reference coefficients were tabulated in aerodecks as functions of mach number and total angle of attack by Zipfel [3], using the aero database Missile DATCOM.

4 Thrust Vector Control

Following the example set by industry, the TVC strategy outlined here was derived with the goal of linearisable control system dynamics. The approach includes a spring-mass-damper actuator model, acceleration controllers relating normal and tangential acceleration to pitch and yaw respectively as well as a roll controller. Each control algorithm was built from the framework described by Zipfel [4], based on the coordinate system and equations of motion outlined above. Note that while the following sections outline the development of the pitch based controller, due to symmetry when considering thrust vector control in isolation, the yaw based controller can be derived using a similar approach. Models to enable thrust vector control for bilaterally symmetric clusters consisting of one, three or n boosters are also presented.

4.1 Single Booster Model

Fig. 5 presents a graphical representation of the coordinate systems used to model a single booster, indicating the engine gimbal angular deflections η and ζ which define the right hand engine coordinate system $]^N$, as well as the body axes 1^B, 2^B, 3^B with rotational rates p (roll), q (pitch), r (yaw).

This simple model allows for the effect of altering the thrust vector direction to be resolved in body coordinates. The transformation matrix required to convert from engine gimbal coordinates to body coordinates is given by:

$$[T]^{NB} = \begin{bmatrix} \cos\zeta\cos\eta & \sin\zeta\cos\eta & -\sin\eta \\ -\sin\zeta & \cos\zeta & 0 \\ \cos\zeta\sin\eta & \sin\zeta\sin\eta & \cos\eta \end{bmatrix} \tag{19}$$

Utilising the transpose of Eq. 19, the thrust vector $[\bar{f}_p]^N = [F \quad 0 \quad 0]$ can be obtained

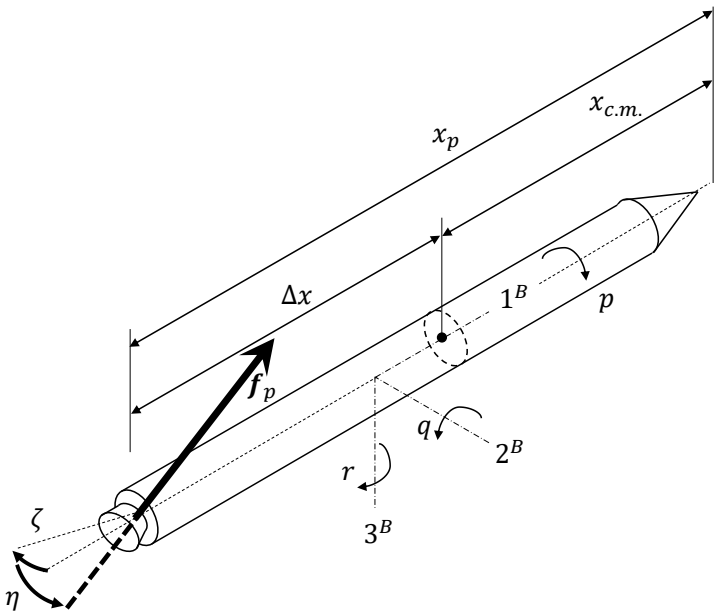

Figure 5: Single booster coordinate system definition, indicating the engine gimbal angular deflections η and ζ as well as the body axes 1^B, 2^B, 3^B with rotations p (roll), q (pitch), r (yaw). Note that the booster image is simply used to clarify the coordinate system definitions, and is not representative of a real booster.

in body coordinates, resulting in the force:

$$\begin{aligned}[f_p]^B &= [\bar{T}]^{NB}[f_p]^N \\ &= \begin{bmatrix} \cos\eta\cos\zeta \\ \cos\eta\sin\zeta \\ -\sin\eta \end{bmatrix} F \end{aligned} \tag{20}$$

As any change in thrust vector direction produces a misalignment between the thrust vector and the 1^B body axis, an expression for the resultant torque also needs to be established. Once the thrust force has been resolved in body coordinates, considering the distance between the engine gimbal and the centre of gravity, Δx, gives the moment

about the centre of gravity:

$$[m_p]^B = \begin{bmatrix} 0 & -(f_p)_3^B & (f_p)_2^B \\ (f_p)_3^B & 0 & -(f_p)_1^B \\ -(f_p)_2^B & (f_p)_1^B & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ 0 \\ 0 \end{bmatrix}$$
$$= \begin{bmatrix} 0 \\ -\sin\eta\Delta x \\ -\cos\eta\sin\zeta\Delta x \end{bmatrix} F \tag{21}$$

Then applying the small angle approximation, $\sin\theta = \theta$ and $\cos\theta = 1$ gives:

$$\delta[m_p]^B = \begin{bmatrix} 0 \\ -\eta\Delta x \\ -\zeta\Delta x \end{bmatrix} F$$

Providing a linear relationship between small engine deflections and the resulting moment increment. The small angle approximation is applicable here since engine deflections are typically limited to approximately $\pm 10°$ by physical constraints. Then defining the control actions for pitch and yaw as δq and δr respectively gives:

$$\delta q = -\delta\eta \tag{22}$$

$$\delta r = -\delta\zeta \tag{23}$$

Directly relating engine gimbal angular deflections to angular rate "deflections" about the body axis. The angular rate deflections are significant as they form the command output of both the acceleration and roll controllers used in this study. Note that as expected, the roll control action (defined as δp) has no effect for a single booster.

4.2 Actuator Control

The model used to represent the actuator dynamics is a second order spring-mass-damper system. The pitch actuator model is illustrated by the sketch and free body diagram (FBD) in Fig. 6, where η_c represents the pitch deflection command and η the actual pitch deflection.

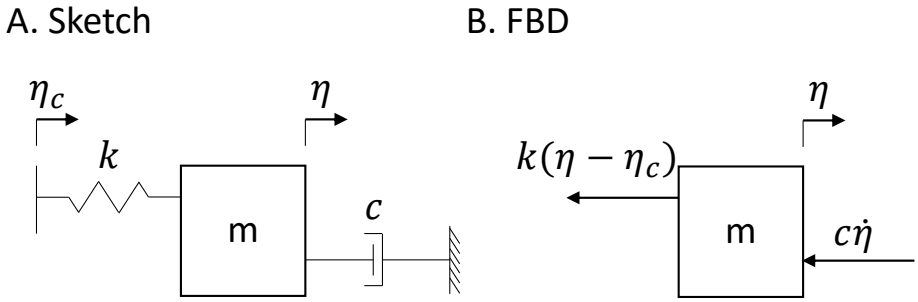

Figure 6: Model of the actuator. A: Sketch of the model; B: Corresponding free body diagram (FBD).

Notably, this spring-mass-damper configuration is the physical representation of a proportional-derivative (PD) controller with derivative action in the return path, resulting in low overshoot when responding to a step input. The step response is relevant as the acceleration controller supplies the actuator with a deflection (step) command at each time step. It is also important to note that the model used for the yaw deflection is obtained by simply replacing the pitch variables η with the yaw variables ζ, so it is excluded for the sake of reducing repetition.

Once the actuator model has been expressed by a FBD (Fig. 6B), the closed loop transfer function can be obtained using classical control techniques. Applying Newton's second law in the Laplace domain gives the following equation:

$$ms^2\eta = -k(\eta - \eta_c) - c\dot{\eta}$$

Then grouping terms and dividing by m gives:

$$\left(s^2 + \frac{c}{m}s + \frac{k}{m}\right)\eta = \frac{k}{m}\eta_c$$

Substituting the natural frequency $\omega_n = \sqrt{\frac{k}{m}}$ and damping ratio $\zeta_d = \frac{c}{2\sqrt{km}}$ leads to the familiar transfer function for a second order system (Eq. 24), allowing for the classical performance paradigms to be applied:

$$\frac{\eta}{\eta_c} = \frac{\omega_n^2}{s^2 + 2\zeta_d\omega_n s + \omega_n^2} \tag{24}$$

Zipfel further tailors the actuator model to the physical system through the addition of deflection and deflection rate limiters. The conversion between the controller deflection command (δq_c) and the actuator deflection command (η_c) using Eq. 22 is embodied in the initial gain, which scales as function of dynamic pressure. This dependence ensures that at low dynamic pressure, the nozzle has high gain and vice versa. The complete block diagram representation of the actuator model is shown in Fig. 7.

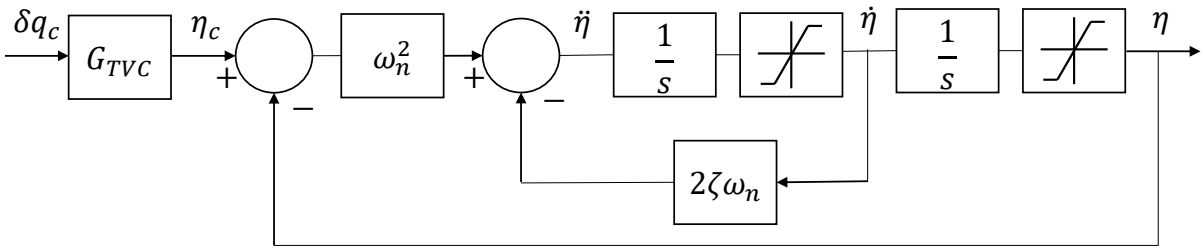

Figure 7: Block diagram of the actuator model.

The major advantage attained through the use of a second order model is access to the classical performance equations for peak, rise (from 10-90%) and settling (to within 2%) times as well as percent overshoot, which are described by Eq. 25-28 respectively. This enables the physical characteristics of the actuator, i.e. the damping ratio ζ_d (not to be confused with the nozzle yaw angle, ζ) and natural frequency ω_n, to be chosen during

design given a set of performance requirements.

$$T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta_d^2}} \tag{25}$$

$$T_r = \frac{2.16\zeta_d + 0.6}{\omega_n}, \qquad 0.3 \leq \zeta_d \leq 0.8 \tag{26}$$

$$T_s = \frac{4}{\zeta_d \omega_n} \tag{27}$$

$$\%OS = \exp \frac{-\zeta_d \pi}{\sqrt{1-\zeta_d^2}} \times 100 \tag{28}$$

4.3 Acceleration Control

The acceleration controller used as the interface between the guidance algorithm and the actuator model was developed based on the linear perturbation equations described in Section 3.2, which form the system plant. It is worth noting that after the design process is complete, the linearised plant is only used implicitly. This is because the result of the controller action (a change in thrust direction when passed through the respective actuator model) is directly incorporated into the full computation of vehicle dynamics. The plant equation for the normal acceleration controller is shown in Eq. 29:

$$\begin{bmatrix} \dot{q} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} M_q & \frac{M_\alpha}{N_\alpha} \\ N_\alpha & -\frac{N_\alpha}{V} \end{bmatrix} \begin{bmatrix} q \\ a \end{bmatrix} + \begin{bmatrix} M_\delta \\ 0 \end{bmatrix} \delta q_c \tag{29}$$

Where q is the pitch rate, δq_c represents the commanded actuator deflection and V the flight speed. The dimensional derivatives of the normal force N and pitching moment M given by:

$$N_\alpha = \frac{\bar{q}S}{m} C_{N_\alpha}; \quad M_\alpha = \frac{\bar{q}Sd}{I_2} C_{m_\alpha}; \quad M_q = \frac{\bar{q}Sd^2}{2I_2 V} C_{m_q}; \quad M_\delta = \frac{-F|x_p - x_{c.m.}|G_{TVC}}{I_2}$$

Where $\bar{q}$ represents the dynamic pressure, S the maximum cross sectional area, d the rocket diameter and I the moment of inertia matrix. F is defined as the thrust magnitude, with $|x_p - x_{c.m.}|$ indicating the thrust vector moment arm about the centre of

mass and G_{TVC} is the gain defined above in Section 4.2. C_{N_α}, C_{m_α} and C_{m_q} are nondimensional derivatives based on the aerodynamic properties of the vehicle. The pitch and yaw symmetry identified for the actuator models exists for the normal and tangential acceleration controllers too, with only the geometric values changing (i.e. the tangential controller considers a different moment of inertia component and use different values for the required coefficients).

The plant equation described by Eq. 29 is representative of a single-input-multiple-output system, with a single control action. Considering this, note that the general state space form of the plant is:

$$\dot{\boldsymbol{x}} = \boldsymbol{F}\boldsymbol{x} + \boldsymbol{g}u \tag{30}$$
$$y = \boldsymbol{C}\boldsymbol{x}$$

Where in this case:

$$\boldsymbol{x} = \begin{bmatrix} q \\ a \end{bmatrix}; \quad \boldsymbol{F} = \begin{bmatrix} M_q & \frac{M_\alpha}{N_\alpha} \\ N_\alpha & -\frac{N_\alpha}{V} \end{bmatrix}; \quad \boldsymbol{g} = \begin{bmatrix} M_\delta \\ 0 \end{bmatrix}; \quad u = \delta q_c; \quad \boldsymbol{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

The block diagram representation of the plant is shown in Fig. 8, with thick lines representing vector pathways.

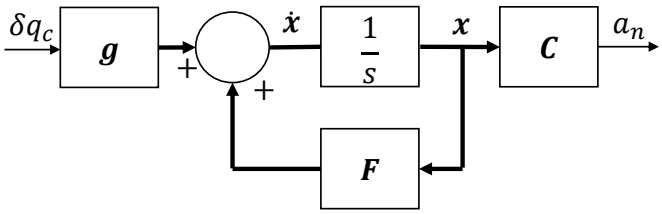

Figure 8: Block diagram of the linear time-variant plant.

While this establishes a relationship between the pitch control deflection and the normal acceleration, the desired input is that of a normal acceleration command. The simple extension is to include a state-feedback controller (analogous to a proportional controller for single-input-single-output, or SISO, systems) as depicted in Fig. 9, where

a "perfect" inertial navigation system (INS) facilitates the feedback pathway.

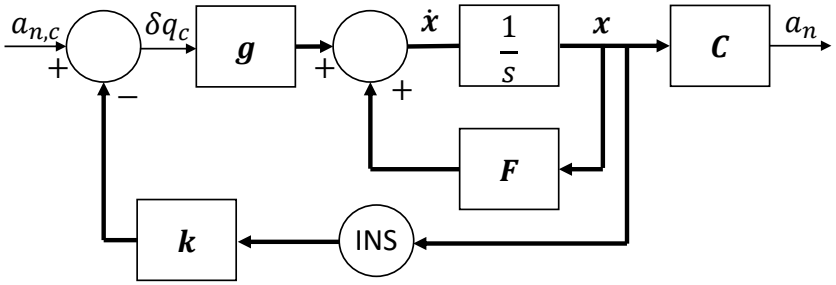

Figure 9: Block diagram of the acceleration control problem.

However, similar to SISO control, state feedback is not sufficient to eliminate steady state error without the assistance of integral action. Thus another state variable, x_N, chosen to be the integral of the acceleration error, is introduced as an auxiliary variable as shown in Fig. 10.

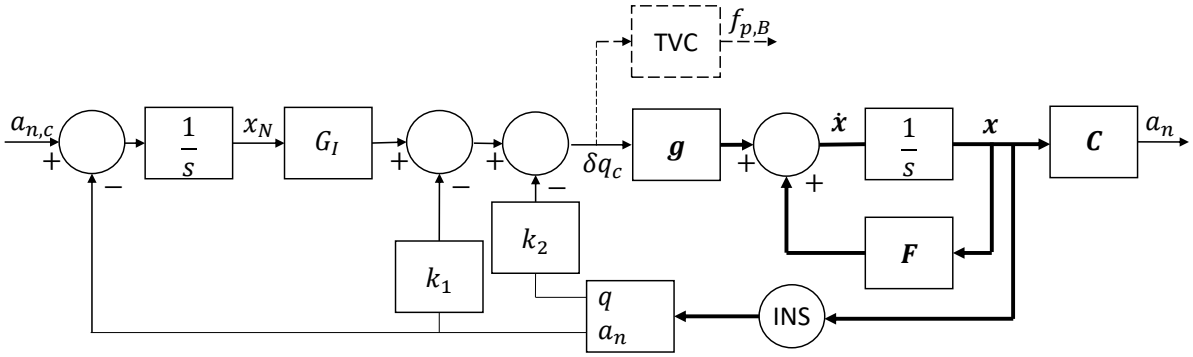

Figure 10: Block diagram of the acceleration controller with integral action.

Then, including the new state variable it can be seen from Fig. 10 that the following relationships hold:

$$\dot{x}_N = -\boldsymbol{C}\boldsymbol{x} + a_c \tag{31}$$

$$\delta q_c = -k_1 a - k_2 q + G_I x_N = -\boldsymbol{k}\boldsymbol{x} + G_I x_N = -\begin{bmatrix} \boldsymbol{k} & -G_I \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ x_N \end{bmatrix} \tag{32}$$

So extending the state space equations (Eq. 30) to include the new state variable:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} & 0 \\ -\boldsymbol{C} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ x_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{g} \\ 0 \end{bmatrix} \delta q_c + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} a_c$$

$$a = \begin{bmatrix} \boldsymbol{C} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ x_N \end{bmatrix}$$

And utilising the expression for δ_{qc} in Eq. 32 gives:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} (\boldsymbol{F} - \boldsymbol{g}\mathbf{k}) & \boldsymbol{g}G_I \\ -\boldsymbol{C} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ x_N \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} a_c \tag{33}$$

$$a = \begin{bmatrix} \boldsymbol{C} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ x_N \end{bmatrix} \tag{34}$$

The transfer function can be obtained by first taking the Laplace transform of Eq. 33 and rearranging for the state space vector:

$$\begin{bmatrix} \boldsymbol{x} \\ x_N \end{bmatrix} = \left(s\boldsymbol{I}_{3\times 3} - \begin{bmatrix} (\boldsymbol{F} - \boldsymbol{g}\mathbf{k}) & \boldsymbol{g}G_I \\ -\boldsymbol{C} & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} a_c$$

where $\boldsymbol{I}$ is the identity matrix, then substituting into the Laplace transform of Eq. 34:

$$\frac{a}{a_c} = \begin{bmatrix} \boldsymbol{C} & 0 \end{bmatrix} \left(s\boldsymbol{I}_{3\times 3} - \begin{bmatrix} (\boldsymbol{F} - \boldsymbol{g}\mathbf{k}) & \boldsymbol{g}G_I \\ -\boldsymbol{C} & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{C} & 0 \end{bmatrix} \boldsymbol{A}^{-1} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \tag{35}$$

Yet to be specified in the acceleration controller transfer function (Eq. 35) is the integral gain (G_I) as well as the state space feedback gains $\boldsymbol{k} = [k_2 \quad k_1]$ for the INS measured pitch rate (q) and acceleration (a). These are computed on-line using the pole placement technique, allowing for the closed loop poles to be scheduled as functions of the dynamic pressure. This is done by setting the system characteristic equation equal to the char-

acteristic equation obtained by choosing n poles, p_i (where n is the order of the state space), and equating coefficients:

$$s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0 = \prod_{i=1}^{n}(s - p_i) \tag{36}$$

The characteristic equation of the system is given by the determinant of the matrix $\boldsymbol{A}$ in Eq. 35. Furthermore, in this case the state space is third order as the system has two states and one integrator, so Eq. 36 reduces to:

$$Det\left|\left(s\boldsymbol{I}_{3\times 3} - \begin{bmatrix} (\boldsymbol{F} - \boldsymbol{g}\mathbf{k}) & \boldsymbol{g}G_I \\ -\boldsymbol{C} & 0 \end{bmatrix}\right)\right| = \prod_{i=1}^{3}(s - p_i)$$

Then expanding gives:

$$Det\left|\begin{bmatrix} (s - M_q + M_\delta k_2) & (-\frac{M_\alpha}{N_\alpha} + M_\delta k_1) & -M_\delta G_I \\ -N_\alpha & (s + \frac{N_\alpha}{V}) & 0 \\ 0 & 1 & s \end{bmatrix}\right| = (s - p_1)(s - p_2)(s - p_3) \tag{37}$$

Which allows for three poles to be chosen, one complex conjugate pair and one real. The "five times" rule of thumb then states that for a third order system if the real pole is atleast five times farther to the left than the dominant complex poles its effect is negligible and the system can be approximated as second order [20]. The corresponding set of poles is:

$$\begin{aligned} p_1 &= -\zeta\omega_n + i\omega_n\sqrt{1 - \zeta_d^2} \\ p_2 &= -\zeta\omega_n - i\omega_n\sqrt{1 - \zeta_d^2} \\ p_3 &= -p \end{aligned}$$

Where p_1 and p_2 are the roots of the mass-spring-damper characteristic equation:

$$s^2 + 2\zeta_d\omega_n s + \omega_n^2 = (s - p_1)(s - p_2) = 0$$

And p is the magnitude of the real pole (not to be confused with the role rate used in other sections of this manuscript). Note that this allows for the classical second order performance equations for peak, rise and settling times as well as percent overshoot (Eq. 25-28 respectively) to be used to specify the damping ratio ζ_d (not to be confused with the nozzle yaw angle) and natural frequency ω_n of the acceleration controller response. To complete the process of specifying the controller performance, Eq. 37 was solved for the controller gains:

$$G_I = \frac{\omega_n^2 p}{N_\alpha M_\delta} \tag{38}$$

$$k_2 = \frac{1}{M_\delta}\left(2\zeta_d\omega_n + p + M_q - \frac{N_\alpha}{V}\right) \tag{39}$$

$$k_3 = \frac{1}{N_\alpha M_\delta}\left(\omega_n^2 + 2\zeta_d\omega_n p + M_\alpha + \frac{M_q N_\alpha}{V} - k_2\frac{M_\delta N_\alpha}{V}\right) \tag{40}$$

A gain scheduling routine such as that described by Eq. 41 and Eq. 42 can be used to adjust the vehicle response as a function of dynamic pressure, $\bar{q}$.

$$\omega_n = 1.5 \times (0.1 + 0.5 \times 10^{-5} \times (\bar{q} - 20 \times 10^3)) \tag{41}$$

$$p = 0.7 + 1.5 \times 10^{-5} \times (\bar{q} - 20 \times 10^3) \tag{42}$$

This is important to ensure that the speed of manoeuvres is limited during the region of high dynamic loading. The damping ratio ζ_d is fixed at a value of 1 to produce a critically damped acceleration response at low dynamic pressures.

4.4 Multiple Booster Model

The single booster model presented in Section 4.1 is extended here to include two additional boosters (see Fig. 11). The configuration was chosen to be representative of the SPARTAN first-stage rocket described in Section 3.3, i.e. a three parallel booster cluster.

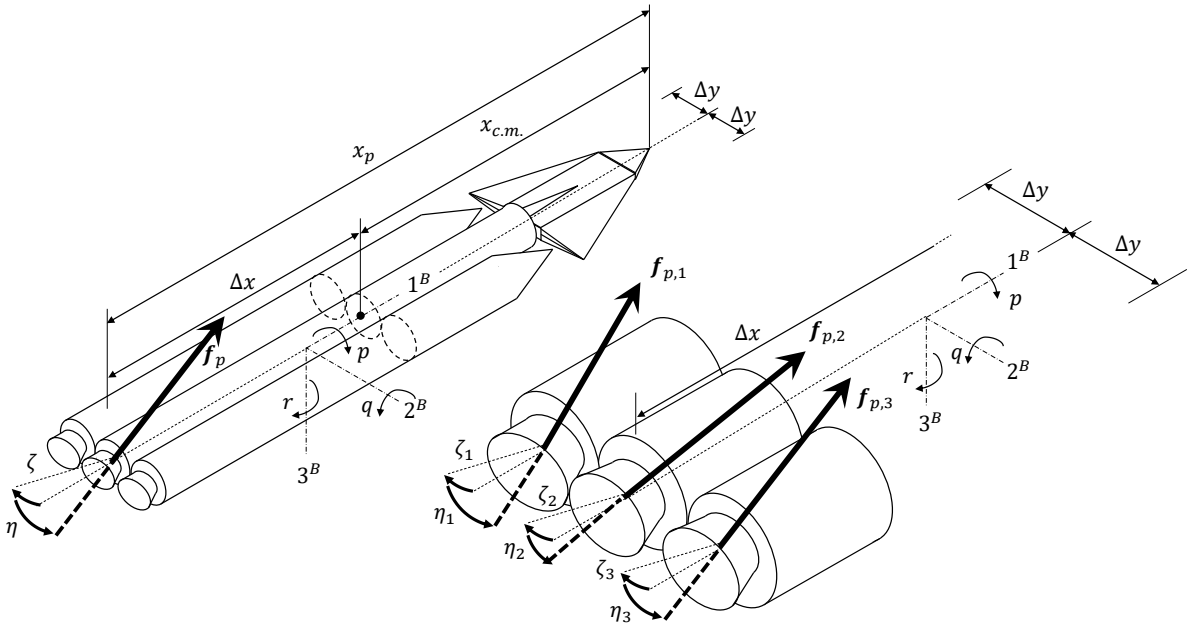

Figure 11: Multiple booster coordinate system definition, defining the engine gimbal angular deflections relative to body coordinates.

The process of converting from engine gimbal coordinates to body coordinates described for the single booster case is repeated here for each additional booster. Specifically, this requires the application of the transformation matrix described by Eq. 19 to convert the three thrust forces $\boldsymbol{f}_{p,1}$, $\boldsymbol{f}_{p,2}$ and $\boldsymbol{f}_{p,3}$ from their respective gimbal coordinates to the global body coordinates. Noting that the thrust of each booster in engine coordinates is $[\bar{f_{p,i}}]^N = [F \quad 0 \quad 0]$, summing the contribution of each booster gives the total thrust force in body coordinates:

$$[f_p]^B = \begin{bmatrix} \cos\eta_1 \cos\zeta_1 + \cos\eta_2 \cos\zeta_2 + \cos\eta_3 \cos\zeta_3 \\ \cos\eta_1 \sin\zeta_1 + \cos\eta_2 \sin\zeta_2 + \cos\eta_3 \sin\zeta_3 \\ -\sin\eta_1 - \sin\eta_2 - \sin\eta_3 \end{bmatrix} F \tag{43}$$

Then considering the moment arm associated with each booster:

$$b_1 = \begin{bmatrix} \Delta x \\ \Delta y \\ 0 \end{bmatrix}, \quad b_2 = \begin{bmatrix} \Delta x \\ 0 \\ 0 \end{bmatrix}, \quad b_3 = \begin{bmatrix} \Delta x \\ -\Delta y \\ 0 \end{bmatrix}$$

The moment about the centre of gravity created by each booster can be obtained by through extension of the relationship described by Eq. 21, resulting in the following contributions:

$$[m_{p,1}]^B = \begin{bmatrix} \sin\eta_1 \Delta y \\ -\sin\eta_1 \Delta x \\ -\cos\eta_1 \sin\zeta_1 \Delta x + \cos\eta_1 \cos\zeta_1 \Delta y \end{bmatrix} F \tag{44}$$

$$[m_{p,2}]^B = \begin{bmatrix} 0 \\ -\sin\eta_2 \Delta x \\ -\cos\eta_2 \sin\zeta_2 \Delta x \end{bmatrix} F \tag{45}$$

$$[m_{p,3}]^B = \begin{bmatrix} -\sin\eta_3 \Delta y \\ -\sin\eta_3 \Delta x \\ -\cos\eta_3 \sin\zeta_3 \Delta x - \cos\eta_3 \cos\zeta_3 \Delta y \end{bmatrix} F \tag{46}$$

Which in turn leads to the total moment of the cluster in body coordinates:

$$[m_p]^B = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} F \tag{47}$$

Where:

$$
\begin{aligned}
M_1 =& (\sin\eta_1 - \sin\eta_3)\Delta y \\
M_2 =& (-\sin\eta_1 - \sin\eta_2 - \sin\eta_3)\Delta x \\
M_3 =& (-\cos\eta_1 \sin\zeta_1 - \cos\eta_2 \sin\zeta_2 - \cos\eta_3 \sin\zeta_3)\Delta x + (\cos\eta_1 \cos\zeta_1 - \cos\eta_3 \cos\zeta_3)\Delta y
\end{aligned}
$$

Then applying the small angle approximation ($\sin\theta = \theta$ and $\cos\theta = 1$) gives a relationship between the moment increment and the engine deflections similar to that of the single booster case:

$$
\delta[m_p]^B = \begin{bmatrix} (\eta_1 - \eta_3)\Delta y \\ (-\eta_1 - \eta_2 - \eta_3)\Delta x \\ (-\zeta_1 - \zeta_2 - \zeta_3)\Delta x \end{bmatrix} F \tag{48}
$$

As before, defining the control actions with the additional inclusion of roll control as δp gives:

$$
\delta p = \frac{1}{2}(\delta\eta_1 - \delta\eta_3) \tag{49}
$$

$$
\delta q = \frac{1}{3}(-\delta\eta_1 - \delta\eta_2 - \delta\eta_3) \tag{50}
$$

$$
\delta r = \frac{1}{3}(-\delta\zeta_1 - \delta\zeta_2 - \delta\zeta_3) \tag{51}
$$

Which conversely results in the engine gimbal angular deflection commands, defined as linear combinations of the pitch, roll and yaw commands:

$$
\begin{aligned}
\delta\eta_1 &= -\delta q + \delta p & \delta\zeta_1 &= -\delta r \\
\delta\eta_2 &= -\delta q & \delta\zeta_2 &= -\delta r \\
\delta\eta_3 &= -\delta q - \delta p & \delta\zeta_3 &= -\delta r
\end{aligned} \tag{52}
$$

4.4.1 n-Booster Generalisation

It can be observed that during the above derivation, the introduction of additional boosters only marginally increased the complexity of the deflection combination relationships (see Eq. 22 and Eq. 23 for the single booster case, and Eq. 52 for the three booster case). While additional boosters inherently impact the structural and aerodynamic considerations of the vehicle, the relationships required for thrust vector control scale in direct proportion to the number of boosters. The generalisation to an n-booster model for thrust vector control is presented here.

Let each booster, i, in the cluster of n boosters have an associated set of engine gimbal coordinates, $]^{N_i}$, with deflection angles η_i and ζ_i, as well as a displacement from the centre of gravity, $\bar{b_i} = [\Delta x_i \quad \Delta y_i \quad \Delta z_i]$. It is assumed that the orientation of each set of engine coordinates is such that when adjusted independently, the angle η_i deflects through the normal plane (i.e. parallel to the plane containing both body axis 1^B and 3^B), and the angle ζ moves through the tangential plane (i.e. parallel to the plane containing both body axis 1^B and 2^B). Furthermore, assume that all boosters within the cluster have the same thrust magnitude, F, with the resultant thrust vector given in each engine frame as $[\bar{f_{p,i}}]^N = [F \quad 0 \quad 0]$. The transformation matrix from any set of engine gimbal coordinates, $]^{N_i}$ to the body coordinates, $]^B$ is:

$$[T]^{N_iB} = \begin{bmatrix} \cos\zeta_i \cos\eta_i & \sin\zeta_i \cos\eta_i & -\sin\eta_i \\ -\sin\zeta_i & \cos\zeta_i & 0 \\ \cos\zeta_i \sin\eta_i & \sin\zeta_i \sin\eta_i & \cos\eta_i \end{bmatrix} \tag{53}$$

So the thrust force contribution from each booster i in body coordinates is:

$$[f_{p,i}]^B = [\bar{T}]^{N_iB}[f_{p,i}]^N = \begin{bmatrix} \cos\eta_i \cos\zeta_i \\ \cos\eta_i \sin\zeta_i \\ -\sin\eta_i \end{bmatrix} F \tag{54}$$

And the total thrust of the cluster in body axis is obtained by summing the contribution of each booster:

$$[f_p]^B = \sum_i^n [f_{p,i}]^B = \sum_i^n \begin{bmatrix} \cos\eta_i \cos\zeta_i \\ \cos\eta_i \sin\zeta_i \\ -\sin\eta_i \end{bmatrix} F \tag{55}$$

Considering the engine gimbal displacements, $\bar{b_i}$ as defined above, the moment about the centre of gravity created by each of the boosters is given by:

$$\begin{aligned} [m_{p,i}]^B &= \begin{bmatrix} 0 & -(f_{p,i})_3^B & (f_{p,i})_2^B \\ (f_{p,i})_3^B & 0 & -(f_{p,i})_1^B \\ -(f_{p,i})_2^B & (f_{p,i})_1^B & 0 \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \end{bmatrix} \\ &= \begin{bmatrix} 0 & \sin\eta_i & \cos\eta_i \sin\zeta_i \\ -\sin\eta_i & 0 & -\cos\eta_i \cos\zeta_i \\ -\cos\eta_i \sin\zeta_i & \cos\eta_i \cos\zeta_i & 0 \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \end{bmatrix} \\ &= \begin{bmatrix} \sin\eta_i \Delta y_i + \cos\eta_i \sin\zeta_i \Delta z_i \\ -\sin\eta_i \Delta x_i - \cos\eta_i \cos\zeta_i \Delta z_i \\ -\cos\eta_i \sin\zeta_i \Delta x_i + \cos\eta_i \cos\zeta_i \Delta y_i \end{bmatrix} \end{aligned} \tag{56}$$

Leading to the total moment of the cluster in body coordinates:

$$[m_p]^B = \sum_i^n [m_{p,i}]^B = \sum_i^n \begin{bmatrix} \sin\eta_i \Delta y_i + \cos\eta_i \sin\zeta_i \Delta z_i \\ -\sin\eta_i \Delta x_i - \cos\eta_i \cos\zeta_i \Delta z_i \\ -\cos\eta_i \sin\zeta_i \Delta x_i + \cos\eta_i \cos\zeta_i \Delta y_i \end{bmatrix} \tag{57}$$

Then applying the small angle approximation ($\sin\theta = \theta$ and $\cos\theta = 1$) provides a relationship between angular deflections and the resultant moment increment as before:

$$\delta[m_p]^B = \sum_i^n \begin{bmatrix} \eta_i \Delta y_i + \zeta_i \Delta z_i \\ -\eta_i \Delta x_i \\ -\zeta_i \Delta x_i \end{bmatrix} \tag{58}$$

Defining the control actions for roll, pitch and yaw as δp, δq, δr respectively gives:

$$\delta p = \frac{1}{k}\sum_{i}^{n}(\text{sign}(\Delta y_i)\delta\eta_i + \text{sign}(\Delta z_i)\delta\zeta_i) \tag{59}$$

$$\delta q = \frac{1}{n}\sum_{i}^{n}(-\delta\eta_i) \tag{60}$$

$$\delta r = \frac{1}{n}\sum_{i}^{n}(-\delta\zeta_i) \tag{61}$$

Where k is the number of boosters with either Δy_i or Δz_i non-zero and the sign function is used to indicate the sign dependence on the direction of the particular displacement in body coordinates. The converse relationships describing the engine gimbal angular deflection commands are:

$$\begin{aligned}
&\delta\eta_i = -\delta q + \text{sign}(\Delta y_i)\delta p && \text{if } \Delta y_i \neq 0\text{; else:}\\
&\delta\eta_i = -\delta q\\
&\delta\zeta_i = -\delta r + \text{sign}(\Delta z_i)\delta p && \text{if } \Delta z_i \neq 0\text{; else:}\\
&\delta\zeta_i = -\delta r
\end{aligned} \tag{62}$$

An important caveat of this model is that all engine gimbal coordinates must be defined such that they are oriented in the same way. This limits the possible booster configurations to those which have bilateral symmetry when viewed axially (i.e. along the 1^B axis).

4.5 Roll Control

The roll controller used in this study was derived using the small perturbation equation for roll:

$$I_{11}\dot{p} = LL_p p + LL_{\delta p}\delta p \tag{63}$$

Expressing Eq. 63 in the Laplace domain then gives the transfer function between the roll control action, δp and the roll rate, p:

$$\frac{p(s)}{\delta p(s)} = \frac{LL_{\delta p}}{s - LL_p} \tag{64}$$

Where $LL_{\delta p} = (\bar{q}Sb/I_{11})C_{l_{\delta p}}$ and $LL_p = (\bar{q}Sb/I_{11})(b/2V)C_{l_p}$ are the dimensioned roll control and damping derivatives respectively (the LL convention is used here as L typically indicates lift). The dynamic pressure is represented by $\bar{q}$, while S indicates the reference area, b the wing span and V the vehicle speed. The roll controller built around this transfer function is described in Fig. 12, featuring two feedback loops with gains K_ϕ and K_p.

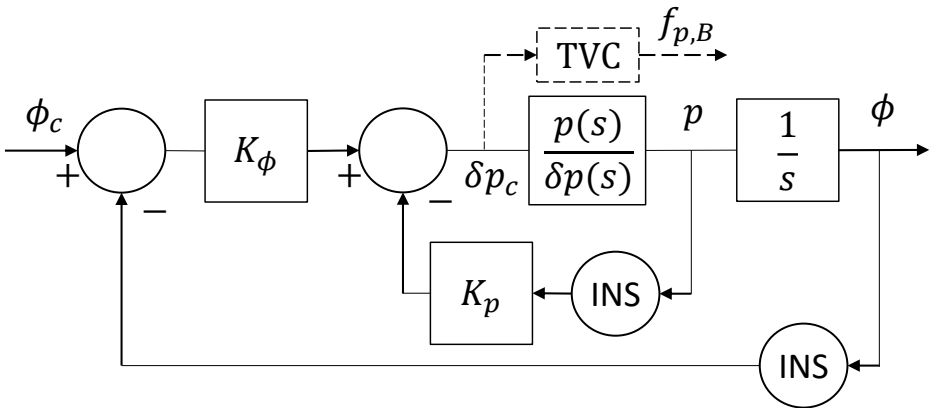

Figure 12: Block diagram of the roll controller model.

The closed-loop transfer function for the roll controller is then obtained using block diagram reduction, giving:

$$\frac{\phi}{\phi_c} = \frac{K_\phi LL_{\delta p}}{s^2 + (K_p LL_{\delta p} - LL_p)s + K_\phi LL_{\delta p}} = \frac{\omega_n^2}{s^2 + 2\zeta_d \omega_n s + \omega_n^2} \tag{65}$$

As the closed-loop transfer function is second order, the gains can be found by equating

coefficients when comparing with the standard form.

$$K_\phi = \frac{\omega_n^2}{LL_{\delta p}} \tag{66}$$

$$K_p = \frac{2\zeta_d \omega_n + LL_p}{LL_{\delta p}} \tag{67}$$

This allows for the system natural frequency, ω_n, and damping ratio, ζ_d (not to be confused with the engine gimbal deflection, ζ) to be fixed using the classical second-order design paradigms (Eq. 25 - Eq. 28))

5 Guidance

The linear tangent guidance law is derived first in planar form using the calculus of variations, then extended for application in 6DOF following the example set by Lawden [26] and Jaggers [28].

5.1 Linear Tangent Guidance

Bryson and Ho [27] demonstrate that optimal control techniques can be used to derive a planar guidance law for fuel optimal orbital insertion by recognising that for constant thrust, obtaining a fuel optimal solution is equivalent to finding a minimum time solution. Following their lead, the linear tangent guidance law is derived here based on the assumption of constant thrust and gravity, as well as no aerodynamics and a flat earth. While restrictive, the affect of these assumptions is reduced when the resultant steering law is iteratively updated. The derivation is initially restricted to planar motion, where the control action is given by a single thrust vector angle, β. For such a system, the equations of motion are given by:

$$\dot{u} = a\cos\beta$$

$$\dot{v} = a\sin\beta - g$$

$$\dot{x} = u$$

$$\dot{y} = v$$

Where a represents the thrust acceleration, g the gravitational acceleration, (x, y) and (u, v) the lateral and vertical displacements and velocities respectively. For convenience,

the above terms are grouped and vector notation is used:

$$\dot{\underline{x}} = f \qquad \text{with} \qquad \underline{x} = \begin{bmatrix} u \\ v \\ x \\ y \end{bmatrix}, \quad f = \begin{bmatrix} a\cos\beta \\ a\sin\beta - g \\ u \\ v \end{bmatrix} \tag{68}$$

To obtain a time optimal solution, set $\phi = 0$ and the Lagrangian, $L = 1$, so that the cost is simply the elapsed time: $J = t_f - t_0$. Then the Hamiltonian is given by:

$$\begin{aligned} H &= L + \lambda^T f \\ &= 1 + \lambda_1 a\cos\beta + \lambda_2(a\sin\beta - g) + \lambda_3 u + \lambda_4 v \end{aligned} \tag{69}$$

And the costate equation is as follows:

$$\dot{\lambda} = -\left(\frac{\partial f}{\partial \underline{x}}\right)^T \lambda - \cancelto{0}{\left(\frac{\partial L}{\partial \underline{x}}\right)^T} = -\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^T \lambda = -\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = -\begin{bmatrix} \lambda_3 \\ \lambda_4 \\ 0 \\ 0 \end{bmatrix}$$

So, solving for each influence function λ_i:

$$\begin{array}{lcl} \dot{\lambda_1} = -\lambda_3 & & \lambda_1 = -\lambda_3 t + c_1 \\ & \text{Integrate} & \\ \dot{\lambda_2} = -\lambda_4 & & \lambda_2 = -\lambda_4 t + c_2 \\ & \text{w.r.t. time} & \\ \dot{\lambda_3} = 0 & & \lambda_3 = c_3 \\ & \rightarrow & \\ \dot{\lambda_3} = 0 & & \lambda_4 = c_4 \end{array}$$

Then the optimality condition (remembering that β is the control variable) is given by the following derivative:

$$\frac{\partial H}{\partial \beta} = 0 \qquad t_0 \le t \le t_f$$

And using the previous expression for the Hamiltonian (Eq. 69) leads to an expression for the control variable:

$$\frac{\partial H}{\partial \beta} = \lambda_1 a \sin\beta + \lambda_2 a \cos\beta = 0$$

$$\rightarrow \tan\beta = \frac{\lambda_2}{\lambda_1} = \frac{-c_4 t + c_2}{-c_3 t + c_1} \tag{70}$$

The time optimal condition provides another equation linking the influence functions:

$$(\phi + H)_{t=t_f} = (H)_{t=t_f} = 0$$

$$\rightarrow (\lambda_1 a \cos\beta + \lambda_2(a\sin\beta - g) + \lambda_3 u + \lambda_4 v)_{t=t_f} = -1 \tag{71}$$

So noting the boundary conditions (initially the rocket is at the origin and stationary, then at a final altitude h and unspecified range $x(T)$ with zero vertical velocity and a horizontal velocity of U):

$$\begin{aligned} u(0) &= 0 & u(T) &= U \\ v(0) &= 0 & v(T) &= 0 \\ x(0) &= 0 & y(T) &= h \\ y(0) &= 0 \end{aligned}$$

And constructing the following relationship to account for the unspecified boundaries:

$$\lambda_i(T) = \begin{cases} \nu_j \\ (\frac{\partial \phi}{\partial x_j})_{t=t_f} \end{cases}$$

Where ν_j is defined such that $u(T) = U$, $v(T) = 0$ and $y(T) = h$, so:

$$\lambda_1 = \nu_1$$

$$\lambda_2 = \nu_2$$

$$\lambda_3 = \frac{\partial \phi}{\partial x}|_{t=T} = 0$$

$$\lambda_4 = \nu_4$$

Then since $\lambda_3 = c_3 = 0$, the expression for the control variable (Eq. 70) becomes:

$$\tan\beta = \frac{-c_4 t + c_2}{c_1} \tag{72}$$

And noting that at time $t = 0$, the control angle is given by initial condition $\beta = \beta_0$, it is evident that:

$$\frac{c_2}{c_1} = \tan\beta_0$$

So setting $c = c_4/c_1$ produces the form of the linear tangent law:

$$\tan\beta = \tan\beta_0 - ct \tag{73}$$

Then solving the equations of motion (Eq. 68) and applying the time optimal condition (Eq. 71) provides an expression for c:

$$c = \frac{a(\tan\beta_0 - \tan\beta_f)\log\left(\frac{\tan\beta_0+\sec\beta_0}{\tan\beta_f+\sec\beta_f}\right)}{U(\tan\beta_0 - \tan\beta_f)} \tag{74}$$

With the required initial and resulting final thrust vector angles (β_0 and β_f respectively) obtained by simultaneously solving:

$$\frac{a}{g} = \frac{\tan\beta_0 - \tan\beta_f}{\sec\beta_0 - \sec\beta_f} \tag{75}$$

$$\frac{2ah}{U^2} = \frac{\tan\beta_0\sec\beta_f - \tan\beta_f\sec\beta_0 - \log\left(\frac{\tan\beta_0+\sec\beta_0}{\tan\beta_f+\sec\beta_f}\right)}{\log\left(\frac{\tan\beta_0+\sec\beta_0}{\tan\beta_f+\sec\beta_f}\right)^2} \tag{76}$$

A vector based approach to the problem was derived by Lawden [26], providing a means of extending the steering law into 3D. The derivation presented here is an adaptation of the approach taken by Robinson [21] in replicating Lawden's result. The

cost function is set to obtain a time optimal solution as in the planar case:

$$J = \phi = t_f - t_0 \tag{77}$$

However in this case the equations of motion are described in vector form:

$$\begin{bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ \frac{F}{m}\hat{\boldsymbol{u}} + \boldsymbol{g} \end{bmatrix} \tag{78}$$

Where $\boldsymbol{r}$ and $\boldsymbol{v}$ represent the displacement and velocity respectively, F the thrust magnitude, m the rocket mass, $\boldsymbol{g}$ the gravitational vector and $\hat{\boldsymbol{u}}$ a unit vector in the thrust direction. Additionally, the conditions on final state are set to be $(\boldsymbol{r}(t_f), \boldsymbol{v}(t_f))$. Robinson [21] sets the Lagrangian to zero for his derivation, relying on the penalty function to provide a solution. The Hamiltonian is given by:

$$H = \begin{bmatrix} \boldsymbol{\lambda}_r \\ \boldsymbol{\lambda}_v \end{bmatrix}^T \begin{bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \boldsymbol{\lambda}_r \cdot \boldsymbol{v} + \boldsymbol{\lambda}_v \cdot \boldsymbol{g} + \boldsymbol{\lambda}_v \cdot \left(\frac{F}{m}\hat{\boldsymbol{u}}\right) \tag{79}$$

Along with the costate equation:

$$\begin{bmatrix} \dot{\boldsymbol{\lambda}}_r \\ \dot{\boldsymbol{\lambda}}_v \end{bmatrix} = -\begin{bmatrix} \frac{\partial H}{\partial \boldsymbol{r}} & \frac{\partial H}{\partial \boldsymbol{v}} \end{bmatrix}^T = \begin{bmatrix} -(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{r}})^T \boldsymbol{\lambda}_v \\ -\boldsymbol{\lambda}_r \end{bmatrix}$$

The transversality condition is then used to incorporate the boundary conditions:

$$\frac{\partial \phi}{\partial \boldsymbol{r}(t_f)} + \boldsymbol{v}^T \cdot \frac{\partial \psi}{\partial \boldsymbol{r}(t_f)} = \boldsymbol{\lambda}_r^T(t_f) \tag{80}$$

$$\frac{\partial \phi}{\partial \boldsymbol{v}(t_f)} + \boldsymbol{v}^T \cdot \frac{\partial \psi}{\partial \boldsymbol{v}(t_f)} = \boldsymbol{\lambda}_v^T(t_f) \tag{81}$$

$$\frac{\partial \phi}{\partial t_f} + v \cdot \frac{\partial \psi}{\partial t_f} + H(t_f) = 0 \tag{82}$$

Applying Pontryagin's principle to maximise the Hamiltonian (Eq. 79) gives the equiva-

lent form:

$$\max_{\hat{\boldsymbol{u}}} \left[\boldsymbol{\lambda}_v \cdot \frac{F}{m} \hat{\boldsymbol{u}} \right] = \lambda_v \frac{F}{m} \tag{83}$$

Which in turn results in the steering law:

$$\boldsymbol{a}_t = \frac{F}{m} \frac{\boldsymbol{\lambda}_v}{\lambda_v} = \frac{F}{m} \hat{\boldsymbol{u}} \tag{84}$$

With $\boldsymbol{a}_t$ representing the thrust acceleration command. Robinson [21] notes that this relationship indicates that the thrust should always be maximised and aligned with the vector $\boldsymbol{\lambda}_v$ for an optimal solution. Assuming constant gravity, the time optimal solution is given by:

$$\boldsymbol{\lambda}_v = \boldsymbol{\lambda}_{v,f} + (t_f - t)\dot{\boldsymbol{\lambda}}_v \tag{85}$$

Where $\boldsymbol{\lambda}_{v,f}$ and $\dot{\boldsymbol{\lambda}}_v$ are constants obtained by solving the boundary value problem presented by the transversality conditions (Eq. 80-82).

Jaggers [28] presents a steering law based on the vector form of the linear tangent guidance scheme derived above:

$$\boldsymbol{u_T} = \frac{\boldsymbol{u_V} + \dot{\boldsymbol{\lambda}}(t - t_\lambda)}{\|\boldsymbol{u_V} + \dot{\boldsymbol{\lambda}}(t - t_\lambda)\|} \tag{86}$$

Where $\boldsymbol{u_T}$ represents a unit vector in the direction of the thrust vector, $\boldsymbol{u_V}$ a unit vector in the direction of velocity-to-go and $\dot{\boldsymbol{\lambda}}$ the turning rate of the thrust vector, which is normal to $\boldsymbol{u_V}$. To utilize the above law, Jaggers solves the equations of motion (Eq. 78) to obtain the relationships described by Eq. 87, 88 and 93. Note that the vector notation used by Zipfel [3] (as described in Section 3.2) is readopted here to clearly indicate the associated coordinate systems and better align with the notation used throughout the study:

$$t_\lambda = J/L \tag{87}$$

$$[\dot{\lambda}]^I = \frac{[r_{go}]^I - S[u_\lambda]^I}{Q - St_\lambda} \tag{88}$$

Where J, L, S and Q represent the following integrals:

$$J = \int_0^{t_{go}} \frac{f_p}{m(t)} t dt = -V_{ex}\left(t_{go} + \tau \log\left(1 - \frac{t_{go}}{\tau}\right)\right) \tag{89}$$

$$L = \int_0^{t_{go}} \frac{f_p}{m(t)} dt = -V_{ex} \log\left(1 - \frac{t_{go}}{\tau}\right) \tag{90}$$

$$S = \int_0^{t_{go}} \int_0^{t} \frac{f_p}{m(t)} dt dt = V_{ex}\left((\tau - t_{go}) \log\left(1 - \frac{t_{go}}{\tau}\right) + t_{go}\right) \tag{91}$$

$$Q = \int_0^{t_{go}} \int_0^{t} \frac{f_p}{m(t)} t dt dt = \tau V_{ex}\left((\tau - t_{go}) \log\left(1 - \frac{t_{go}}{\tau}\right) + t_{go}\right) - \frac{1}{2} V_{ex} t_{go}^2 \tag{92}$$

With the characteristic time of the rocket defined as $\tau = \frac{m_0}{\dot{m}_f}$ along with the mass at time t, $m(t) = m_0 - \dot{m}_f t$. Also, by definition:

$$[u_\lambda]^I = \frac{[v_{go}]^I}{\|[v_{go}]^I\|} \tag{93}$$

Each of the above quantities ultimately depend on an accurate estimate of both the time-to-go (t_{go}), velocity-to-go (v_{go}) and range-to-go (r_{go}). Velocity and range-to-go are obtained by comparing the current state with the desired end state using a predictor-corrector scheme. This allows for the time-to-go to be estimated by setting the L integral equal to the velocity-to-go. Although the model used as a basis for the derivation of the linear tangent guidance law is quite simple, the effect of the unaccounted-for phenomenon (i.e. aerodynamic loadings) is minimised through iterative application. Furthermore, in his implementation, Zipfel [3] includes a number of measures to correct for the effects of gravity on the velocity and range-to-go estimates as well as a thrust bias correction routine.

6 Simulation

An overview of the simulation architecture is presented, along with results for both the thrust vector control and guidance modules presented in Sections 4 and 5. Note that the fully characterised generic Small Launch Vehicle [3] is used as the underlying vehicle model for all simulations.

6.1 Architecture

The open source software package CADAC++, developed by Zipfel [1], forms the framework used for building simulations throughout this study. In particular, the Rocket6 simulation [3] is used as a foundation. All modules deemed outside the scope of this study were decommissioned to remove potential sources of error within the results, effectively applying the "ideal sensors" assumption made during the problem statement. An overview of the final structure is described by Fig. 13.

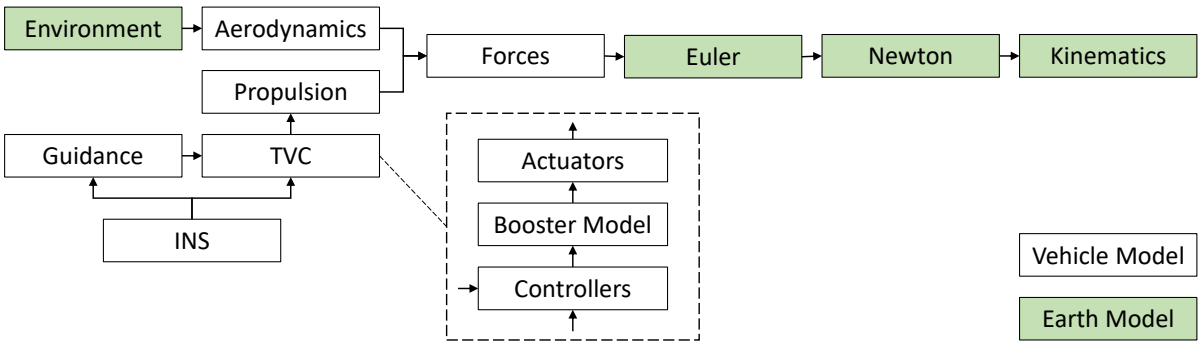

Figure 13: Graphical overview of the CADAC++ simulation architecture [1].

The role of each module highlighted in Fig. 13 is as follows:

1. INS - Ideal inertial navigation system (INS) sensor feedback.

2. Guidance - Implementation of the final variation of the linear tangent guidance routine outlined in Section 5.

3. TVC - Consists of multiple classes providing the acceleration and roll controllers

outlined in Sections 4.3 and 4.5 respectively, as well as the booster and actuator models described in Sections 4.1, 4.4 and 4.2.

4. Propulsion - Updates vehicle properties due to fuel expenditure and sets the thrust magnitude.

5. Environment - Provides the atmospheric (US 1976 Standard Atmosphere) and gravitational (WGS84 Ellipsoid Earth) models.

6. Aerodynamics - Computes aerodynamic forces as well as moment and control coefficients based on input aerodecks.

7. Forces - Computes total force due to dynamic pressure and thrust as well as the aerodynamic moments.

8. Euler - Updates angular rates in body and inertial coordinates.

9. Newton - Updates acceleration, velocity, position and flight path angles.

10. Kinematics - Updates simulation time and Euler angles.

CADAC++ also includes a number of background support modules which are not described here. For details on the simulation package in its entirety see the supporting publication by Zipfel [1].

6.2 Results

Simulation results are presented demonstrating the performance of the thrust vector control schemes outlined in Section 4, and describing an unsuccessful attempt at implementing the linear tangent guidance algorithm.

6.2.1 Thrust Vector Control

A series of simulations were completed to demonstrate the performance of the thrust vector control scheme, including both the single and multiple booster models, as well as the acceleration and roll controllers. Fig. 14 illustrates the performance of the normal acceleration controller described in Section 4.3 for the single booster model. The commanded normal acceleration ($a_{n,c}$) is shown in blue, with a -0.15 g acceleration step command occurring at 16s. The corresponding engine gimbal angular deflection command for the single booster model, η, is portrayed in red, and the normal acceleration response (a_n) in green.

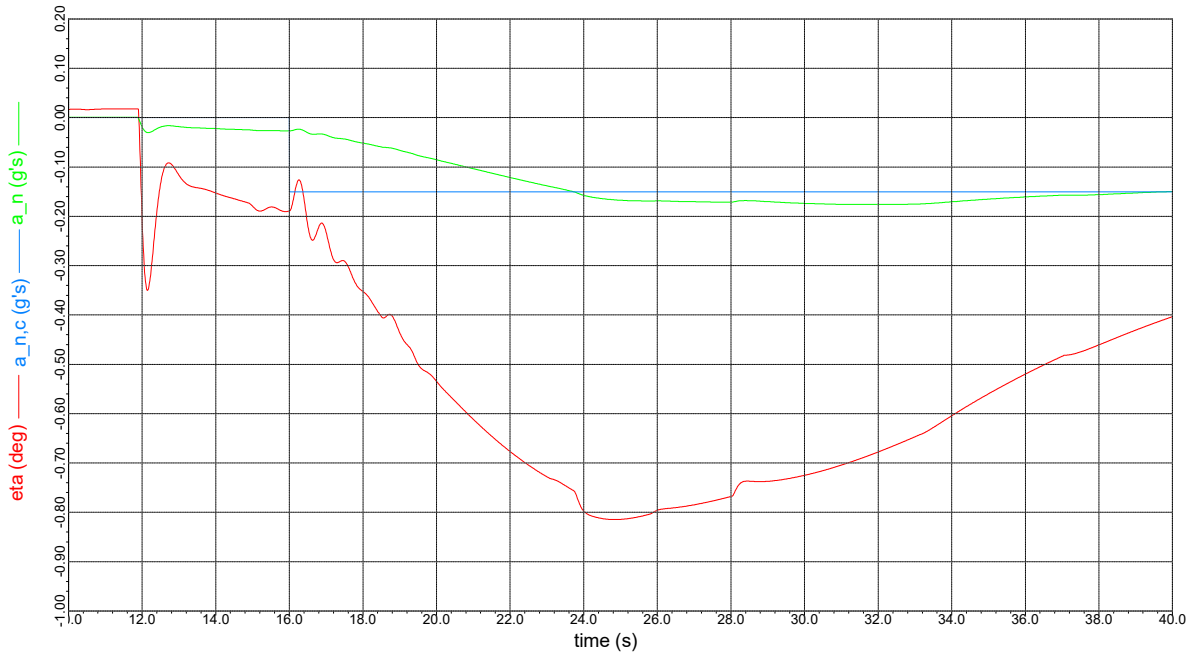

Figure 14: Normal acceleration response, a_n (green) to a step command of $a_{n,c} = -0.15$ g's (blue). The engine gimbal deflection angle, η, in degrees is shown in red.

The performance of both the multiple booster model and the roll controller is

illustrated in Fig. 15. The command sequence consisted of:

1. An initial requirement of zero roll, normal and tangential acceleration
2. A roll step of $+1^\circ$ at 12s
3. A roll step of -1° back to hold at zero in combination with a normal acceleration step of -0.15 g's at 16s.

The top subplot shows the roll step command (ϕ_c) in red, the subsequent roll deflection command (δp_c) in green and the roll response (ϕ) in blue. The corresponding engine gimbal deflections for each η_i are illustrated in the lower subplot.

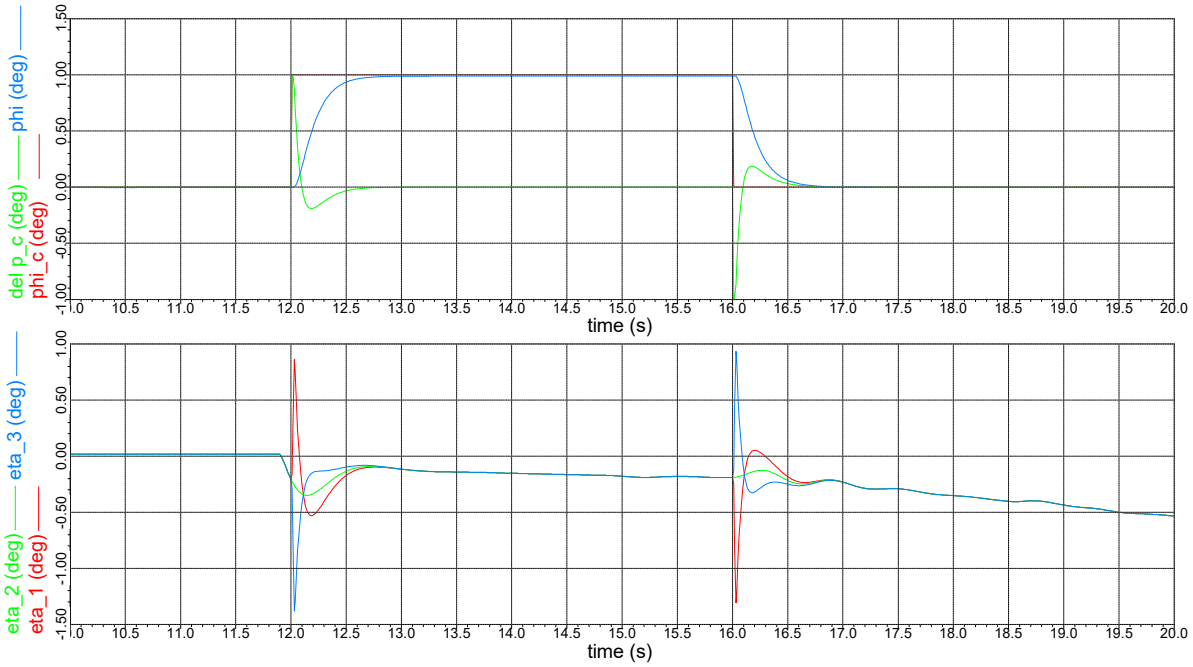

Figure 15: Top: Roll angle response, ϕ (blue) to a roll step command of $\phi_c = 1^\circ$ (red) along with the intermediate roll deflection command, δp_c (green). Bottom: The three engine gimbal η_i deflections.

The pitch and yaw deflection commands (δq_c and δq_c respectively) issued to stabilise normal and tangential accelerations during the roll step manoeuvre are shown in the top subplot of Fig. 16, along with the roll step (ϕ_c) for reference. For completeness, the motion of the lateral engine gimbal angles ζ_i are described within the bottom subplot of that figure.

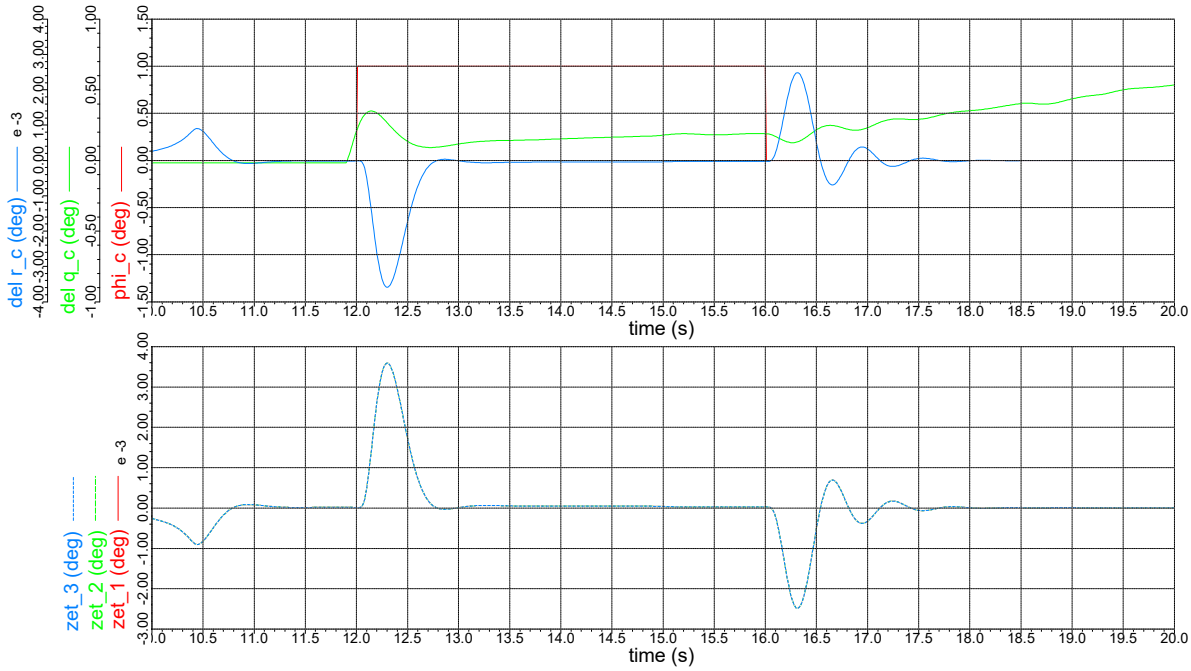

Figure 16: Top: Pitch (δq_c) and yaw (δr_c) deflection command response (green and blue respectively) when subject to a roll step command of $\phi_c = 1°$ (red). Bottom: The three engine gimbal ζ_i deflections.

In order to highlight the performance issues that arise due to restrictively modelling the physical system using position and rate limiters, Fig. 17 shows the response of the multiple booster system to an unrealistic roll step of 120°. Another shortfall of the model is highlighted in Fig. 18 as the result of a 150° roll step. Both of those figures follow the same layout as the earlier roll response plot, with the top subplots showing the roll command (ϕ_c) in red, the subsequent roll deflection command (δp_c) in green and the roll angle response (ϕ) in blue. Similarly, the lower subplot illustrates the corresponding engine deflections (η_i).

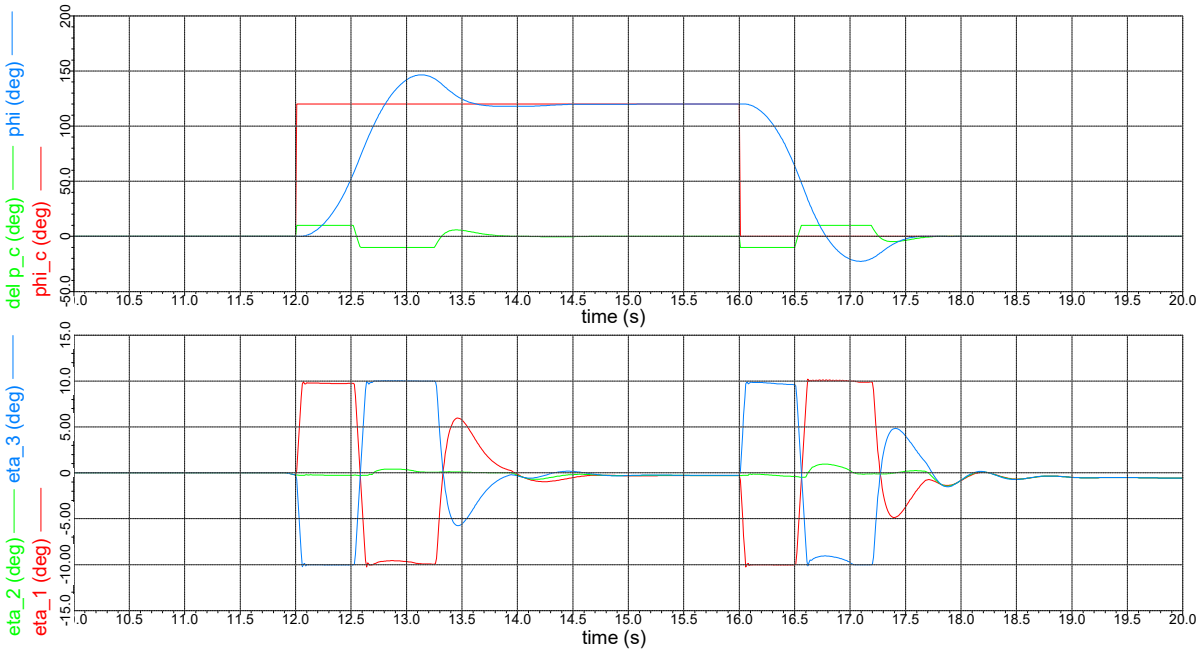

Figure 17: Roll step command of $\phi_c = 120°$. Top: Roll command ϕ_c (red), roll deflection command δp_c (green) and roll response ϕ (blue). Bottom: The three engine gimbal η deflections.

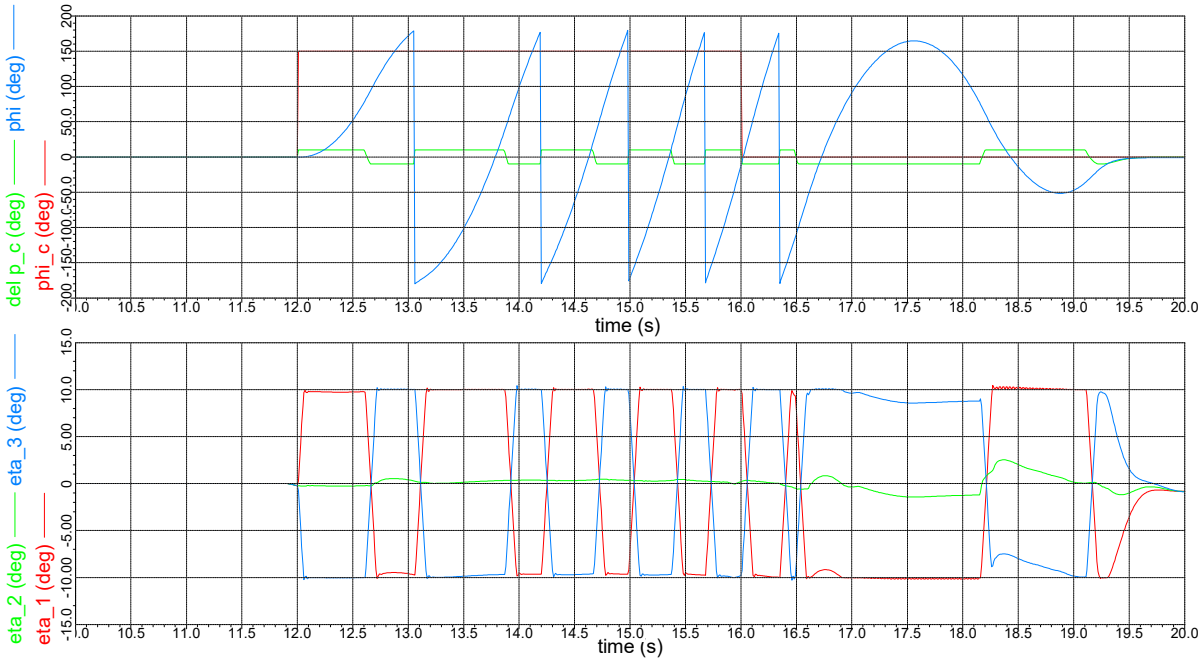

Figure 18: Roll step command of $\phi_c = 150°$. Top: Roll command ϕ_c (red), roll deflection command δp_c (green) and roll response ϕ (blue). Bottom: The three engine gimbal η deflections.

6.2.2 Guidance

The linear tangent guidance algorithm was not successfully implemented for a single stage launch in CADAC++. The approach consisted of modifying the three-stage Rocket6 guidance simulation presented by Zipfel [3], noting that a pitch program rather than a guidance scheme was applied to the first stage in that study. Additionally, Zipfel links the linear tangent guidance command to the second and third-stage reaction control systems in place of the thrust vector control approach utilized here. The null result of the simulation attempt is presented to identify shortcomings of the approach. Immediately evident is the convergence failure of the time and velocity-to-go as shown in Fig. 19. Furthermore, that figure highlights the significant saturation of the thrust turning rate vector throughout the flight (a performance metric suggested by Zipfel [3]).

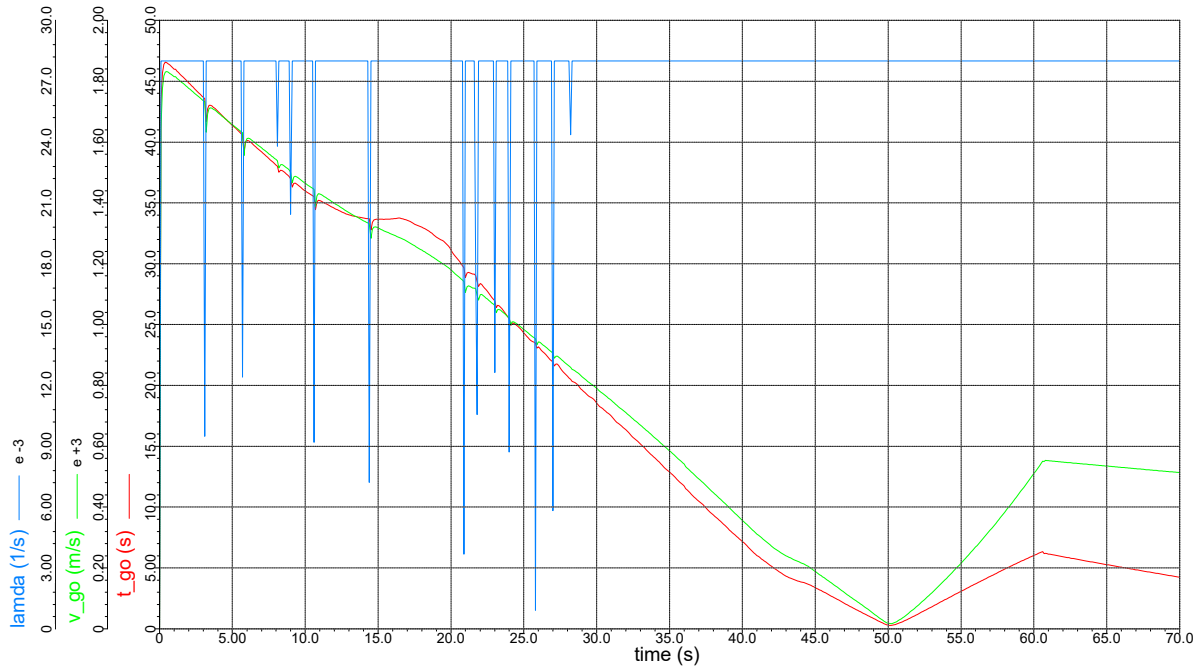

Figure 19: Magnitude of the thrust turning rate vector $\dot{\lambda}$ (blue), magnitude of the velocity-to-go estimate, v_{go} (green) and the time-to-go estimate, t_{go} (red).

In order to investigate the problem effectively, the convergence requirements were relaxed such that the simulation would break once t_{go} was was within 0.5s rather than the 0.05s used by Zipfel. The navigation command, chosen to be somewhat similar to the SPARTAN first-stage boost engine cut-off state (BECO) specified in Tab. 1, as well

as the simulation BECO results are listed in Tab. 5.

Parameter	Orbital Position (m)	Velocity (m/s)	Flight path angle (°)
Command	6400000	1300	75
BECO State	6399610	1288.62	73.69
Error	386.747	11.38	1.31

Table 5: Guidance simulation null result, describing the Small Launch Vehicle boost engine cut-off state following the relaxation of convergence requirements. Mean radius of the Earth for reference when considering orbital position: 6370987.308m.

The corresponding traces for altitude, inertial velocity and flight path angle are shown in Fig. 20A, noting that BECO occurred at $t = 49.6s$. The fuel mass is tracked in Fig. 20B, along with the vehicle mach number and dynamic pressure.

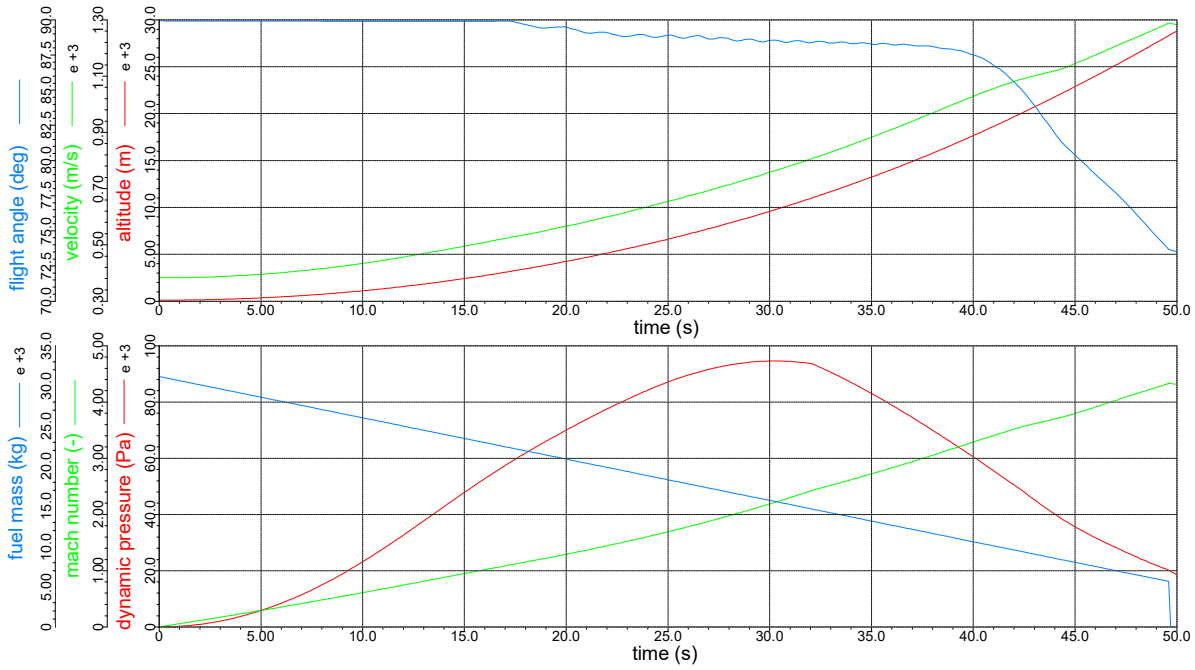

Figure 20: Top: Flight path angle (blue), absolute inertial velocity (green) and vehicle altitude (red). Bottom: Remaining fuel mass (blue), vehicle mach number (green) and dynamic pressure (red).

Identified in Section 5.1 as two of the fundamental parameters required for implementation of the linear tangent guidance algorithm, the time and velocity-to-go are shown in Fig. 21A. Also described in that subplot is the characteristic time, τ, used during the integral computations and representing the hypothetical burn time to zero mass. The individual components of the velocity-to-go vector are shown in Fig. 21B.

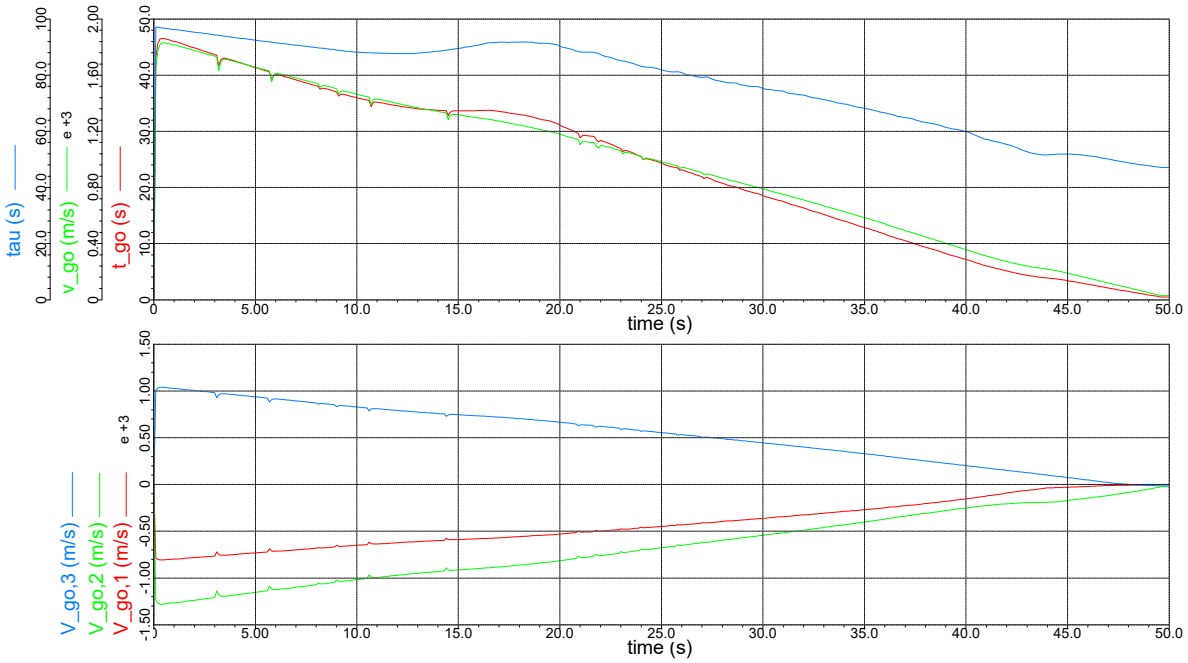

Figure 21: Top: Characteristic time, τ (blue), magnitude of the velocity-to-go estimate, v_{go} (green) and the time-to-go estimate, t_{go} (red). Bottom: Velocity-to-go vector components, $[v_{go}]_i^I$.

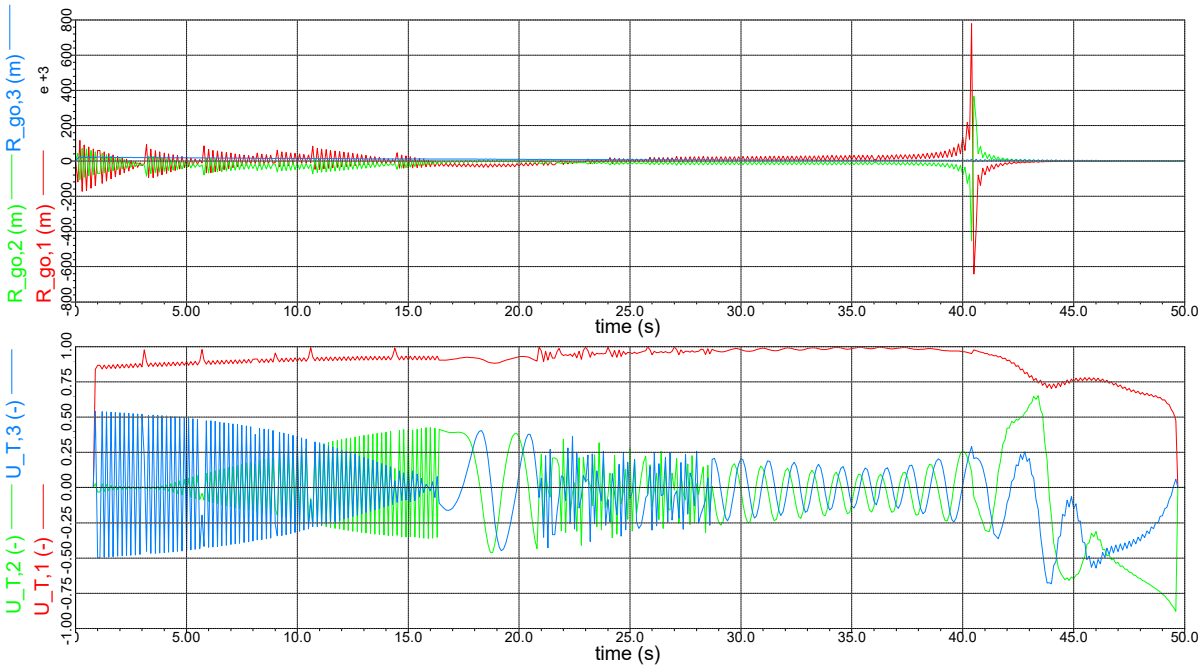

Figure 22: Top: Range-to-go vector components, $[r_{go}]_i^I$. Bottom: Unit thrust vector command components, $[U_T]_i^B$.

The final fundamental quantity required by the linear tangent guidance algorithm, namely the range-to-go vector estimation, is described in Fig. 22A. Notably, this figure introduces the first sign of the underlying erroneous behaviour. The resultant guidance unit thrust vector command signal is described in Fig. 22B.

7 Discussion

An evaluation of the thrust vector control results is presented, and the null result of the attempted guidance scheme implementation is analysed. Suggestions are made regarding the simulation architecture for future guidance implementation attempts, and limitations of the multiple booster thrust vector control model are identified.

7.1 Thrust Vector Control

The results of a performance evaluation completed on the normal acceleration step test shown in Fig. 14, as well as both the 1° and 120° roll step tests shown in Fig. 15A and Fig. 17A respectively are listed in Tab. 6. The rise time is defined as the time taken to progress from 10% to 90% of the command, while the settling time is defined as the time taken to settle to within 2% of the command.

Command	Rise Time (s)	Settling Time (s)	Percent Overshoot (%)
a_n Step (-0.15 g's)	6.4	22.5	20
Roll Step ($+1°$)	0.36	-	0
Roll Step ($-1°$)	0.36	-	0
Roll Step ($+120°$)	0.48	1.52	23.3
Roll Step ($-120°$)	0.45	1.48	20

Table 6: Thrust vector controller performance evaluation.

The slow transient performance of acceleration controller is highlighted in the first row of Tab. 6, with the large time scale also emphasised in Fig. 14. The slow response of the system is understood to be an artefact of the gain scheduling routine which acts as a function of dynamic pressure (see Eq. 41 and Eq. 42). Fig. 23 describes the dynamic pressure experienced during the first-stage boost in blue, as well as the scheduled acceleration controller natural frequency and real pole location in green and red respectively. Note that the damping ratio of the controller is fixed at 1 to ensure a critically damped response at low dynamic pressures. The cost of this rough adjustment is evident by the 20% overshoot, however for a more refined simulation, a higher fidelity gain scheduling scheme would likely provide a significant performance improvement.

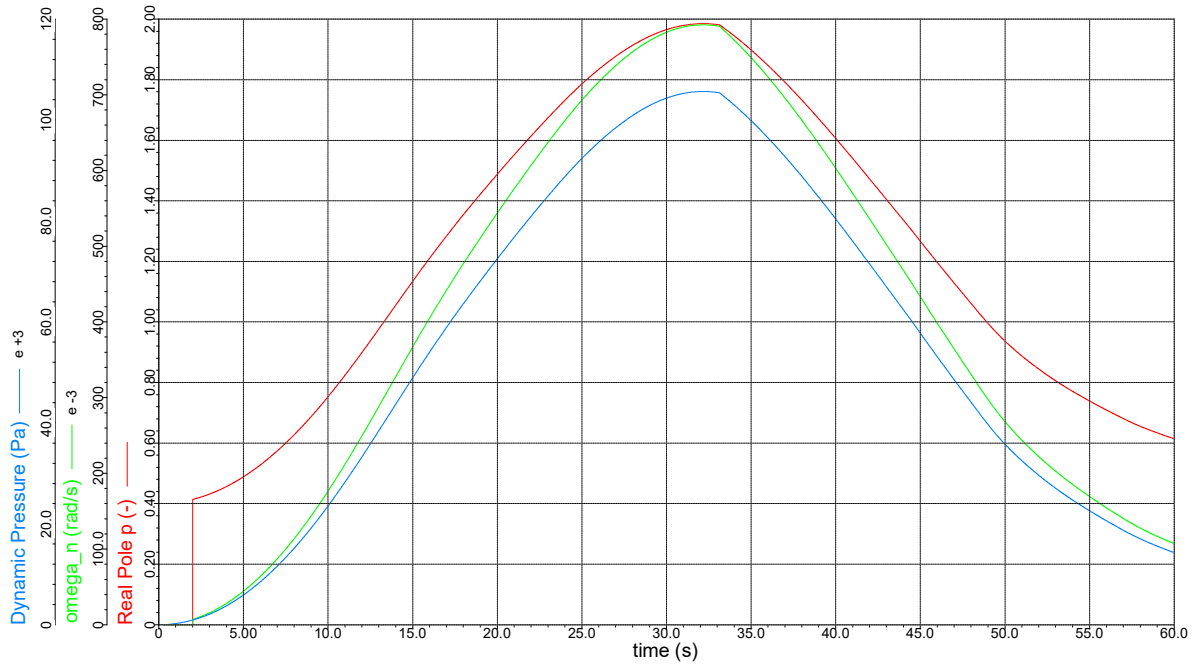

Figure 23: Dynamic pressure, $\bar{q}$ (blue), acceleration controller natural frequency, ω_n (green) and real pole location, p (red) during the first stage boost of normal acceleration step command of $a_{n,c} = -0.15$ g's simulation.

In contrast, the roll controller produced a fast transient response for the 1° step command as highlighted by the rise time of 0.36s. The roll response was critically damped as mandated by design and illustrated in Fig. 15A. Note for a critically damped response, it is impossible to compute the settling time by definition. Although a roll step is an unlikely manoeuvre to intentionally complete during a typical launch scenario, this behaviour is important to avoid unwanted oscillations in the system as a result of disturbances while the roll angle is held at zero. The intermediate step between the command (ϕ_c) and response (ϕ), namely the roll deflection command (δp_c), is shown in green in Fig. 15A. Qualitatively, the behaviour is as expected, with an initial spike initiating the spin of the vehicle followed by a smaller negative peak, slowing the movement to asymptotically approach the commanded roll angle. Of particular note with regard to the roll controller performance is the step down (or $-1°$) manoeuvre initiated at 16s, as this also coincides with a -0.15 g's normal acceleration command. The transient response of the roll angle for that section of the simulation is identical to that of the initial step, with a

rise time of 0.36s. The auxiliary result shown in Fig. 16A demonstrates the pitch and yaw commands (in green and blue respectively) issued to maintain zero normal and tangential acceleration during the roll manoeuvre, followed by the -0.15g's normal acceleration step at 16s. Fig. 16B describes the engine deflection through each of the ζ_i angles, with each engine tracking together. This is expected, as in the chosen three booster configuration deflecting any of the engines through the ζ_i angles solely impacts the vehicle's yaw with no control authority over roll.

A set of simulations with unrealistic roll commands were completed to emphasise the limitations of the controller, and explore the subsequent effects on the system performance. For the 120° case illustrated in Fig. 17, the transient characteristics are described in the bottom two rows of Tab. 6. The immediately evident repercussion of the large step command is that the critically damped response design specification has broken down. Inspection of Fig. 17A indicates that this is due to the 10° limiter placed on the roll deflection command. The consequence of saturating the deflection command is that both the initial impulse observed to rotate the vehicle in the 1° case and the opposing deflection required to slow the motion down are not possible. Rather, smaller insufficient deflection commands are issued, leading to the sluggish response observed – increasing the rise time and introducing overshoot. The limitation on the roll deflection is incorporated to represent the physical limitations of the thrust vector control system, as such the actual value of 10° is vehicle dependent. The simulation conducted with a 150° roll step highlights another shortfall of the model, despite also being an unrealistically large manoeuvre. As can be observed in Fig. 18, the CADAC++ implementation of the model is unable to process roll angles above 180°. This is considered to be an implementation issue which could be fixed by also considering the absolute rotation from the origin. Fortunately, as the typical application of the roll controller during a first stage rocket launch is to maintain a roll angle of zero, neither of these issues have an immediate impact.

In each of the simulations described above, including those in which the roll con-

troller displayed erroneous behaviour, it can be observed that the multiple booster model performed as intended. This is evident through a comparison of Fig. 15A and Fig. 15B, noting that the outer engines (subscripts 1 and 3) deflect in opposing directions through their respective η_i angles when subject to the initial roll deflection impulse, then switch directions in response to the negative command. Furthermore, once the roll manoeuvre had been completed, all three engines track together to enable the vehicle to respond to any pitch (normal acceleration) disturbances or commands. This is emphasised through the response of the engine deflections to the -0.15 g's normal acceleration command at 16s. These observations can also be made for the simulations shown in Fig. 17 and Fig. 18, however the deflections in those cases are limited by the roll deflection saturation.

7.2 Guidance

As noted in Section 6.2.2, a functional linear tangent guidance simulation was not achieved. The three-stage CADAC++ module, Rocket6, presented by Zipfel [3] was adapted for a single-stage, with the guidance command output linked to the thrust vector control (TVC) multiple booster model through conversion of the unit thrust vector into normal and tangential acceleration commands. An analysis of the null results is presented to identify the shortfalls of the attempted implementation, culminating in recommendations for future work.

The traces illustrated for both velocity and altitude in Fig. 20 indicate that an approximate trajectory solution is still obtained following the relaxation of the convergence requirements, a statement confirmed by the BECO state results listed in Tab. 5. However, both the flight path angle oscillations and general lack of asymptotic behaviour indicate that the solution is not optimal. It is important to note that while the total fuel mass of the first stage forms an upper limit on the total flight time, the linear tangent guidance scheme is ideally time optimal, so a non-zero fuel mass at BECO is expected for preliminary simulations.

Two of the fundamental inputs to the guidance scheme, time and velocity-to-go,

are shown in Fig. 21A. For a functional guidance scheme, these values are expected to decay in a linear fashion. The behaviour exhibited in Fig. 21A suggest that this requirement is satisfied. It is worth noting that the time and velocity-to-go are not completely independent, as the time-to-go is obtained through setting the L integral (Eq. 90) equal to the velocity-to-go estimation and solving the resulting function of time. The velocity-to-go is obtained using the predictor-corrector scheme, however the smooth, convergent component traces described in Fig. 21B suggest that this computation is completed without the introduction of significant error.

The primary source of error is evident when observing the range-to-go trace illustrated in Fig. 22A. As the third fundamental input to the guidance process outlined in Section 5.1, the oscillations in the range-to-go estimation propagate through the steering law solution. This leads to the extreme noise seen in the guidance command (see Fig. 22B), with initial large oscillations of the 2nd and 3rd vector components about zero, followed by smaller asymmetric oscillations after a simulation time of approximately 16s. Note that an ideal unit thrust vector guidance command features the 1st component at almost unity, with the 2nd and 3rd components only slightly deviating from zero in either direction.

Following the attempted conversion of the multi-stage code for single-stage use, it became evident that the introduction of the initially small noise levels associated with the range-to-go estimation caused unfavourable feedback throughout the future steering law solutions, dramatically increasing the output error. This was identified as a result of the circular dependence that exists when updating the thrust vector turning rate, $\dot{\lambda}$, and the range-to-go estimation for consecutive time steps.

Fundamental to that process, it is postulated that the initial source of error may be due to the approach of computing integrals, which for the simulation attempt presented here was based on the implementation suggested by Zipfel [3]. That methodology was based on an unattainable internal NASA document, and includes a variation of the predictor-corrector scheme outlined by Jaggers [28]. The required integral formulation

for the multi-stage case features a number of approximations designed to minimise the computation cost of evaluating the integrals, an important consideration at the time of publication. Unfortunately a number of additional integrals were included within that variation of the predictor-corrector scheme, and as the underlying mathematical formulation was not disclosed, the correct adaptation for a single stage was not possible. An outline of the approach recommended for future work, which attempts to remedy the shortfalls identified here, is presented in Section 7.3.

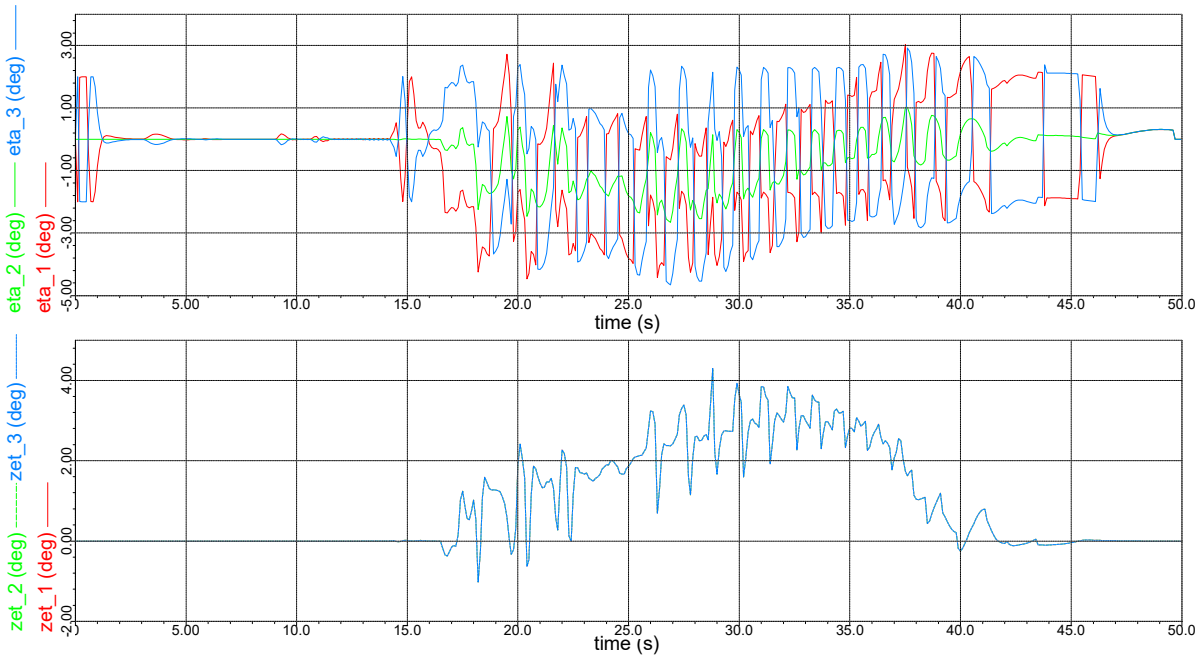

Figure 24: Top: The three engine gimbal η_i deflections. Bottom: The three engine gimbal ζ_i deflections.

The engine gimbal deflection response of the multiple-booster thrust vector control model is illustrated in Fig. 24, with the top and bottom sub-plots describing the η_i and ζ_i angular deflections respectively. Observing the response characteristics of the model when provided with a noisy, erroneous signal such as that exhibited by the guidance unit thrust vector command shown in Fig. 22B yields insight into the relationship between the guidance and TVC modules. The extreme, large oscillations extant in the initial $\sim$ 16s of the guidance command signal causes a loss of vehicle control in the sense that it is unable to respond correctly, however does not result in an equally oscillatory

set of engine gimbal angular displacements. Rather, the three engine gimbal displacements approximate the mean guidance command for that symmetric section of the signal, namely zero. This artificial filtering process is due to the slower response time of the acceleration controllers as identified in Section 7.1, evidently a desirable characteristic with regard to guidance command signal noise. The remainder of the flight features significant η_i oscillations, notably with the 1st and 3rd engines deflecting in alternate directions. These motions highlight the roll controller working to maintain the commanded zero roll angle, despite the otherwise (still) violent vectoring commands from the guidance module. Note that the general trend for the η_i gimbal angles (described exactly by the η_2 trace, as that engine is not effected by the roll controller commands) does coincide with the change in flight angle observed in Fig. 20.

7.3 Recommendations - Improved Guidance Architecture

The following subsection outlines the recommended program flow for the linear tangent guidance module. Note, that the implementation is suggested as a potential solution to the issues identified above and is untested. Each of the following steps should form a function call:

1. Velocity-to-go. The velocity-to-go estimate should be obtained by first combining the boost engine cut-off (BECO) state flight angle and desired velocity magnitude to obtain a representation of the final velocity vector, then combining that vector with the current state velocity.

2. Time-to-go. The time-to-go should be computed by setting the L-integral equal to the magnitude of the velocity-to-go estimate and solving for time. The resultant equation is given by:
$$t_{go} = \tau\left(1 - \exp\left(\frac{-\|V_{go}\|}{V_{ex}}\right)\right) \tag{94}$$

3. Integrals. The integrals defined in Eq. 89-92 should be solved using the analytic solutions provided.

4. Range-to-go. The range-to-go should be computed based on an estimate of the final state using the equations of motion as described by Eq. 78. The most appropriate form of numerical integration for the problem has yet to be investigated.

5. Turning rate. The turning rate, $[\dot{\lambda}]^I$, should be evaluated using the relationship defined by Eq. 88.

6. Thrust vector command. Finally, the unit thrust vector command, $[U_T]^I$, should be computed using Eq, 86.

The relationships between these functions is described in Fig. 25, indicating the required inputs and outputs for each case.

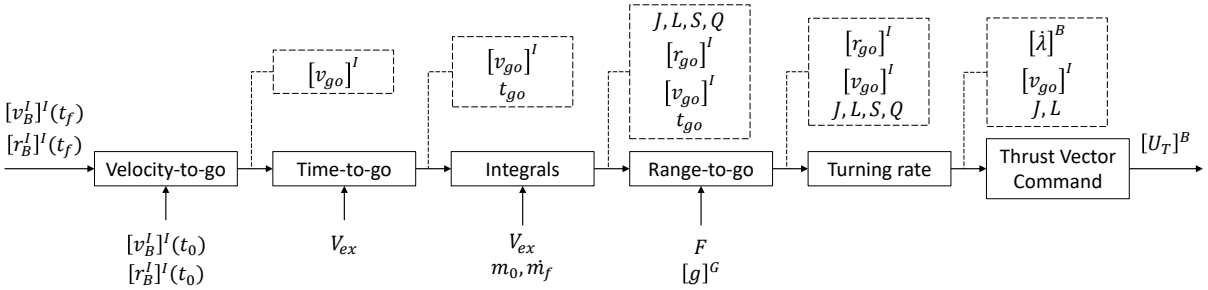

Figure 25: Recommended guidance architecture for future work.

A further consideration that should be made during future implementation attempts is with regard to the point raised by Jaggers [28] – the dependency of linear tangent guidance on "to-go" estimates inherently results in instabilities as t_{go} approaches zero. Although not addressed in this study due to the implementation errors encountered before it became relevant, the recommendation made in that paper, to simply not use linear tangent guidance in the last $\sim$ 4s of the trajectory, should be considered.

7.4 Limitations

A number of limitations were identified throughout the project when considering the thrust vector control (TVC) scheme and multiple booster modelling approach. The first of note was the realisation that the use of acceleration controllers without the additional

inclusion of body or flight angle controllers removes the ability angles to be specified directly, rather providing only implicit angular control. A set of explicit angular controllers could be constructed using an approach similar to that undertaken during the acceleration controller derivation, as outlined in Section 4.3, but instead based on the appropriate set of linear perturbation equations for the angle under consideration. Secondly, the extent of the possible control authority is limited, when considering vehicle manoeuvrability, through the exclusion of body angle rate controllers. The suggestion for solving the previous limitation is applicable for this case also. Finally, as noted in Section 4.4.1, the generalisation of the multiple booster model limits booster configurations to those which have bilateral symmetry when viewed axially (i.e. along the 1^B axis). This last limitation is fundamental to that modelling methodology, and as such, alternative approaches must be used for non-compliant cluster configurations.

8 Conclusion

This study was conducted with the goal of providing a comprehensive modelling approach for the guidance of a first-stage rocket to a specified boost engine cut-off state using thrust vector control. This has been achieved through:

1. The presentation of the underlying theory required to develop a high fidelity multiple booster thrust vector control scheme, as well as the linear tangent guidance law from first principles.

2. An outline of the architecture and mathematical modelling paradigms used to simulate a first-stage launch in 6 degrees of freedom.

3. Simulation results supporting the multiple booster model as well as the thrust vector control regime.

Although issues were encountered with the linear tangent guidance simulation attempts, a suggested methodology for implementing the guidance module is presented in Section 7.3. Future work should address this shortfall, and also investigate the incorporation of body angle and body angular rate controllers into the thrust vector control scheme.

9 Acknowledgements

The authors would like to thank Prof Ross McAree and Dr Michael Kearney for technical advice provided throughout the project.

References

[1] P. H. Zipfel, "Cadac: multi-use architecture for constructive aerospace simulations," *The Journal of Defense Modeling and Simulation*, vol. 9, no. 2, pp. 129–145, 2012.

[2] D. Preller and M. Smart, "Spartan: Scramjet powered accelerator for reusable technology advancement," in *Proceedings of the 12th Reinventing Space Conference*. Springer, 2017, pp. 139–147.

[3] P. H. Zipfel, "Orbital insertion control of a three-stage solid rocket booster modeled in six degrees-of-freedom," *Journal of Modeling, Simulation, Identification, and Control*, vol. 2, no. 1, pp. 31–44, 2014.

[4] ——, *Modeling and simulation of aerospace vehicle dynamics*. Aiaa, 2000.

[5] S. X. Elon musks mission to mars – a merlin engine. [Online]. Available: https://www.wired.com/images_blogs/wiredscience/2012/10/ff_musk2_f.jpg

[6] R. Lab. Rocket lab reveals first battery-powered rocket for commercial launches to space. [Online]. Available: https://www.rocketlabusa.com/latest/rocket-lab-reveals-first-battery-powered-rocket-for-commercial-launches-to-space/

[7] J. Chai and M. Smart, "Gravimetric model of a fly back booster concept based on microcosm boosters," University of Queensland, Centre for Hypersonics, Tech. Rep., 04 2017.

[8] F. J. Gomez and R. Miikkulainen, "Active guidance for a finless rocket using neuroevolution," in *Genetic and Evolutionary Computation Conference*. Springer, 2003, pp. 2084–2095.

[9] G. P. Sutton and O. Biblarz, *Rocket propulsion elements*. John Wiley & Sons, 2016.

[10] E. Seedhouse, *SpaceX: making commercial spaceflight a reality*. Springer Science & Business Media, 2013.

[11] R. Lab. (2016) Paylod user's guide. [Online]. Available: https://www.rocketlabusa.com/assets/Uploads/Payload-User-Guide.pdf

[12] J. S. Orr, J. H. Wall, T. S. VanZwieten, and C. E. Hall, "Space launch system ascent flight control design," 2014.

[13] C. B. Ensworth, *Thrust Vector Control for Nuclear Thermal Rockets.* National Aeronautics and Space Administration, Glenn Research Center, 2013.

[14] E. N. Johnson, A. J. Calise, and J. E. Corban, "Adaptive guidance and control for autonomous launch vehicles," in *Aerospace Conference, 2001, IEEE Proceedings.*, vol. 6. IEEE, 2001, pp. 2669–2682.

[15] Y. Shtessel, C. Hall, and M. Jackson, "Reusable launch vehicle control in multiple-time-scale sliding modes," *Journal of Guidance Control and Dynamics*, vol. 23, no. 6, pp. 1013–1020, 2000.

[16] B. Yu and W. Shu, "A novel control approach for a thrust vector system with an electromechanical actuator," *IEEE Access*, vol. 5, pp. 15 542–15 550, 2017.

[17] D. E. Schinstock, D. A. Scott, and T. A. Haskew, "Modeling and estimation for electromechanical thrust vector control of rocket engines," *Journal of Propulsion and Power*, vol. 14, no. 4, pp. 440–446, 1998.

[18] D. V. Lazić and M. R. Ristanović, "Electrohydraulic thrust vector control of twin rocket engines with position feedback via angular transducers," *Control Engineering Practice*, vol. 15, no. 5, pp. 583–594, 2007.

[19] Y. Li, H. Lu, S. Tian, Z. Jiao, and J.-T. Chen, "Posture control of electromechanical-actuator-based thrust vector system for aircraft engine," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 9, pp. 3561–3571, 2012.

[20] N. S. Nise, *Control Systems Engineering.* John Wiley & Sons, 2007.

[21] S. B. Robinson, "Spacecraft guidance techniques for maximizing mission success," 2014.

[22] B. Suresh and K. Sivan, *Integrated Design for Space Transportation System*. Springer, 2015.

[23] F. H. Martin, "Closed-loop near-optimum steering for a class of space missions," *AIAA Journal*, vol. 4, no. 11, pp. 1920–1927, 1966.

[24] R. H. Battin, ""space guidance evolution–a personal narrative,"," *AIAA Journal of Guidance and Control*, vol. 5, pp. 97–110, 1982.

[25] F. Teren, "Explicit guidance equations for multistage boost trajectories," 1966.

[26] D. F. Lawden, "Rocket trajectory optimization-1950-1963," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 4, pp. 705–711, 1991.

[27] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[28] R. Jaggers, "An explicit solution to the exoatmospheric powered flight guidance and trajectory optimization problem for rocket propelled vehicles," *AIAA paper*, vol. 1051, 1977.

[29] S. K. Sinha and S. K. Shrivastava, "Optimal explicit guidance of multistage launch vehicle along three-dimensional trajectory," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 3, pp. 394–403, 1990.

[30] N. Sarzi-Amade, T. P. Bauer, J. R. Wertz, and M. Rufer, "Sprite, a very low-cost launch vehicle for small satellites," in *Proceedings of the 12th Reinventing Space Conference*. Springer, 2017, pp. 165–178.